

QL Today

Volume 11
Issue 5
June-August
2007

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...



XII

5 Minutes to 12?

No!

**3 Months to
Volume 12!!**

www.QLToday.com

Contents

3	Editorial	
4	News	
7	Electronic QL Today	Geoff Wicks
10	Archivers Cover Disk	Dilwyn Jones
12	Review of QL Wiki	Gerard Phelan
15	Working Space	George Gwilt
26	A Weekend of Waffles?	Geoff Wicks
31	Programming Assembler - Part 18a	Norman Dunbar
32	Letter-Box	
33	ZIP and UNZIP - Part 2	Dilwyn Jones
41	Essential QL Software	David Denham
44	Lightning Strikes Again	Simon N Goodwin
50	Light Relief with Strings	George Gwilt
52	Electro-Magnetic-Pulses	Stephen Poole
54	Byts of Wood	Roy Wood

Advertisers

in alphabetical order

Jochen Merz Software	19
QBranch	34, 35
Quanta	9
RWAP	27
TF Services	45
Geoff Wicks	39

Please fill in the renewal form enclosed if you have not done it already!
Thanks a lot for your continuing support!

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
 Kaiser-Wilhelm-Str. 302 Fax +49 203 502012
 47169 Duisburg email: JMerz@j-m-s.com
 Germany email: QLToday@j-m-s.com

English office:

Q Branch Tel. +44 1273 386030
 20 Locks Hill Mobile +44 7836 745501
 Portslade Fax +44 1273 381577
 BN41 2LB email: qbranch@qbranch.demon.co.uk
 United Kingdom email: QLToday@j-m-s.com

Editor:

Geoff Wicks Tel. +44 1332 271366
 5b Wordsworth Avenue email: gwicks@beeb.net
 Sinfon email: QLToday@j-m-s.com
 Derby DE24 9HQ
 United Kingdom

Co-Editor:

Bruce Nicholls Tel +44 20 71930539
 38 Derham Gardens Fax +44 870 0568755
 Upminster email: qltoday@q-v-d.demon.co.uk
 Essex RM14 3HA email: QLToday@j-m-s.com
 United Kingdom

QL Today is published five times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Pictures may be in .SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2007 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Jonathan Hudsons web site where you find more information about lots of interesting QDOS software and INFOZIP at www.bigfoot.com/~jrudson/

The deadline for the next issue is the
15th of August 2007

Editorial

by Geoff Wicks

Time for a little cheating. Last year we had difficulty in maintaining our publishing schedule, and only produced four issues of QL Today instead of five. As this is the issue where we look at what we achieved during the previous year we have extended 2006 by another month to give us five issues. During 2006 we produced 286 A4 pages, an average of 57 pages per issue. Editorial content was 82%, which comes to 47 pages per issue. We had an average of 4 pages of news per issue, and produced 1 cover disk. 16 different writers contributed articles to QL Today, and a further two people sent letters.

It is worth repeating something we wrote in the last issue;

"Editorially the magazine continues to remain viable and we are lucky to have a group of enthusiastic writers with different levels of QL use and experience."

To each of our writers a grateful thank you with the hope that you will continue to contribute during volume 12. Thanks are also due to all the members of the QL Today team, who do much work invisible to the reader. Jochen continues to have the heaviest burden. A single article, particularly one containing a lengthy basic listing, can take hours to lay out. Roy continues to maintain the UK office and plays an active role in collecting feedback and advising on the style and content of the magazine. Bruce faithfully proof-reads the entire content at very short notice.

A special word of thanks to ex-editor Dilwyn Jones, who continues to write for us and provides much of the content for our cover disks. I have valued his support throughout my time as editor.

Last year we were able to publish hardware articles in three of the five issues although most of these were, strictly speaking, off topic. We hope this will stimulate the original authors or others to contribute QL related hardware articles.

News coverage remains a problem. Only three people consistently send us news items and instead we have to actively seek out news stories. Nowadays there is little "hard" news in terms of new products and developments and increasingly the news content is "soft". We hope readers do not experience this as a "dumbing down" of our news content.

Finally a few words about the "other" publication. The Quanta Magazine has had a good 2006 and can now claim to be a serious QL publication once again. Other good Quanta news is a completely renewed website. Quanta's weakness is its patchy coverage of QL news in general and Quanta news in particular. QL Today believes in healthy competition and we would like to see the Quanta committee set up a news service, both in print and on the web, to rival that of ours.

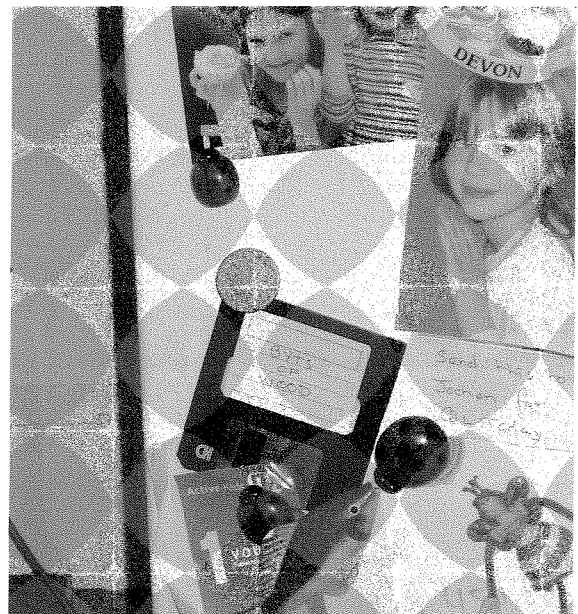
Cartoon

by Roy Wood

The eagle eyed amongst you – and in fact those of you of a more myopic disposition – may have noticed that last issue's Byts of Wood was the same as the previous one.

This was entirely my fault because I emailed the wrong file to Jochen. The one in this issue is the one that should have been in the last issue so some references to things in the magazine are actually referring to things in the last issue. Confused? – you should be.

Anyhow I will endeavour not to do this again and I have started a new system to make sure it does not happen – as you can see...



New Website

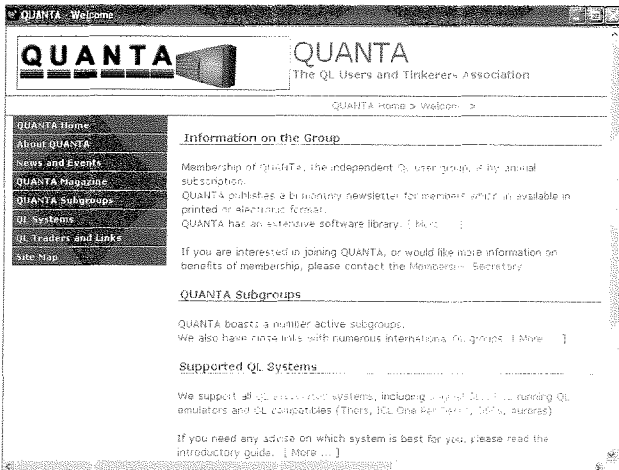
At the beginning of May Quanta launched its new website. A major feature of the site is simplicity of navigation with minimal scrolling necessary.

The new site has a restful colour scheme with black text on a white background. Headings are in blue and links in green. On the left side of the screen is the main menu with white letters on a blue background. At the bottom of the page are links to trader and other sites.

The main pages are Home; About Quanta; News and Events; Quanta Magazine; Sub-groups; QL Systems; Traders and Links; and Site Map. Clicking one of these items will sometimes bring up a submenu to take the user directly to the section he wishes to read.

Users who are interested in seeing an electronic version of the Quanta Magazine can download the February/March 2006 edition. There are also extensive pages on QL hardware. Links to other QL sites are not just on the dedicated page, but also at relevant places within other web pages.

The site was designed by Quanta's new webmaster, **Don Abbott**, who came into the QL community via the OPD. Don had already given a much appreciated preview of the site at the Hove show and AGM. During his presentation he made it clear that he took a flexible view of the site and that he was prepared to modify and update it in response to members' opinions and possible future developments.



Don Abbott has told QL Today that he hopes the new website will not only function as a resource for members, but also help in recruiting new members. He would also like to see Quanta's subgroups maintain their own content. At the moment **Malcolm Cadman** hosts a page for the

London subgroup and **John Sadler** for the Scottish subgroup. There are large subgroups in Manchester and West Midlands as well as smaller ones in Dorset, Surrey, Sussex and the Solent.
www.quanta.org.uk

QL WIKI

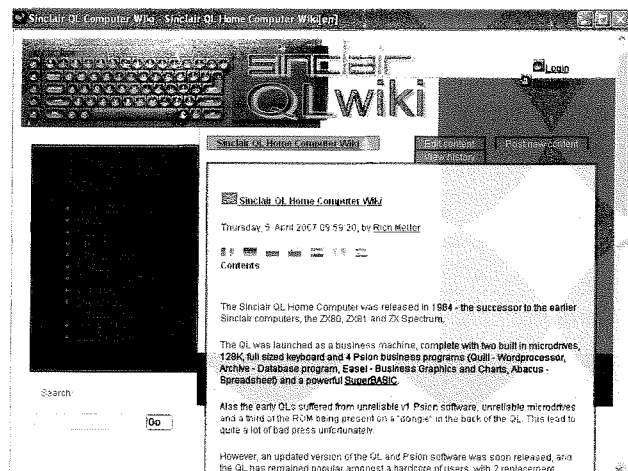
Rich Mellor has launched a QL Wiki at:
http://www.rwapadventures.com/ql_wiki/

Originally intended as a software wiki, the site has now expanded into a general QL Wiki. Topics include QL CD-ROMs; hardware; history; operating systems; QL websites; platforms; software; SuperBasic; and traders and personalities.

Although predominantly an English language site, provision has been made for a number of languages. At the time of writing the opening page is available in Dutch and French. A small amount of the other content has also been translated into French.

Like all wiki sites the content can be edited by the users, although the last news was that only five people have contributed. This means there are some large gaps in the content. There was some discussion on the QL-users group about whether a specific QL Wiki was necessary or if it was better for QL-ers to contribute to the QL entries in the official Wikipedia. Some users argued that by using the latter, we could reach a wider audience. Others pointed out that we could include more information on our own wiki and added that the new site was coming up well on the search engines.

A review of this site appears elsewhere in this issue.



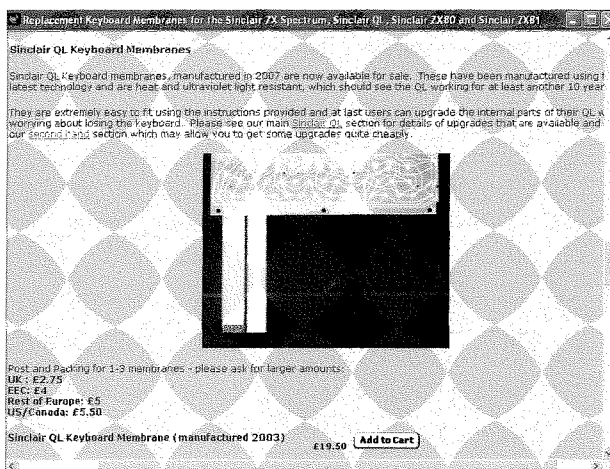
RWAP Move

RWAP has moved. The new contact details are:
3 Dale View Court,
Fulford,
Stoke-On-Trent,
Staffordshire
ST11 9BA
TEL: 01782 398143
E-Mail: rich@rwapservices.co.uk

Keyboard Membranes and More

Quanta has financed by means of a loan the manufacture of 200 new keyboard membranes. This was the smallest order the manufacturers would accept. The membranes are available from RWAP at:

<http://www.rwapsoftware.co.uk/membranes.html>



At the recent Quanta AGM chairman **John Mason** announced this was only one of the items that Quanta was considering financing.

Quanta can purchase the QMAC licence for £100, but purchase of the source code would be more difficult as the costs would be out of proportion to the use that would be made of it. There would be an initial search fee of probably £2,000 to locate the code.

(Readers who wish to remind themselves about QMAC can read the QL Today review in Volume 2 issue 1 page 33)

John Mason indicated that Quanta would also consider sponsoring a software author for a possible USB device. However the committee would first need more information on the feasibility of such a device and would also expect a detailed business plan.

QL Documentation CD

Dilwyn Jones announces a new version of the QL Documentation CD: (QL Today readers received an earlier version with the magazine in 2005): "Version 2.90 of the QL Documentation CD is

now available. As it is essentially freeware, you should be able to get copies from your local QL software traders soon. A copy has been sent to Quanta for members to order through the Quanta librarian (please enquire to the Quanta librarian for costs etc.).

Changes to this CD in the last few months include:

v2.86 (21/06/06) - Corrections to the QL graphics file formats document GRAPHICS.doc in Graphics-> folder.

Added new article about differences between QL screens PIC files and PSA files. Document in Graphics-> folder.

v2.87 (06/08/06) - Easel graphics file article added.

v2.88 (25/01/07) - Qubide manuals and sources added. QEPIII manual added.

v2.89 (22/02/07) - Turbo and TurboPTR articles added.

v2.90 (29/03/07) - Level 2 Upgrade ROM replacement manual added.

Some of the documents are available from my website at

<http://dilwynjones.topcities.com/qldocs/qldocs.html>

although limited space there means that only a selection of the more popular documents are there, usually in one file format only, whereas the CD contains copies of most in a choice of QL text or doc files and Windows DOC, RTF or PDF documents."

QL Web Documentation

Dilwyn Jones writes:

"Having got tired of having my QL offerings spread across two sizeable websites, I have now begun the task of consolidating it all on one site. The first step is to put as much QL documentation as possible onto my main website, which I have started to do at the following web page:

<http://www.dilwyn.uk6.net/docs/idx.html>

From now on, the original documentation page on the TopCities website will not be updated - any new material will go on this page. Once migration of the site is complete (it is larger and more time consuming to transfer than I had thought), everything will have been consolidated on the uk6.net site and the TopCities pages will be closed down eventually.

In addition to the advantage of consolidating everything on one site, the uk6.net ISP has given me "unlimited" space on my main website. Quite how they define "unlimited" is unclear to me, but at least it does mean that given the relatively small size of QL software compared to the bloatware of other operating systems,

"unlimited" is likely to mean "pretty unlimited" to me as a QLer.

It does also mean that I can put pretty well all of the content of the QL Documentation CD online now. Even some of the larger and less used documents like the Thor technical manual can now be put online, albeit at a rate dictated by the fact that I am not on broadband.

If anyone has relevant QL documentation or useful articles which you would like to be placed online, please get in touch with me via email at dilwyn.jones@tesco.net



Other Web Updates

ABACUS – EXCEL Transfers

Just Words! has updated the help file on its help and advice page on transferring spreadsheets from Abacus to a PC. This now includes additional help for people wishing to import a spreadsheet into Excel. *Just Words!* is grateful to **Colin McKay** for writing this additional help.

<http://members.lycos.co.uk/geoffwicks/justwords.htm>

LEVEL 2 Upgrade ROM Manual

Dilwyn Jones writes:

"Following recent discussion of the Level 2 Upgrade ROM for SuperQBoards and Trump Cards on the QL Users mailing list, I have been given permission to put a replacement manual on my website. The Level 2 Upgrade ROM image is not included (it is copyright Jochen Hassler).

The upgrade ROM provides level 2 directories on SuperQBoards and Trump Cards. It includes the ATR device driver for reading Atari and DOS format disks.

Please note that it is not quite the same as the

level 2 ROM built into Qubbesoft Trump Cards – that ROM does not include an ATR device driver.

<http://dilwynjones.topcities.com/qldocs/qldocs.html>

Scroll down to the bottom of the page to click on the download link.

In time, this will be added to the QL Documentation CD in a wider choice of formats than I'm able to include on my website."

UNIVERSAL CONFIG

George Gwilt writes:

"Introducing his "Config for C" Jonathan Hudson said: "How nice, you might think, to use a freely distributable, widely available, standard configuration program to configure your 'C' programs."

The problem of adding Config Blocks to C programs, Assembler programs and S*BASIC programs has at long last been solved.

The program UCONFIG (standing for Universal CONFIG) is now available on the SQLUG site

www.jms1.supanet.com

This program produces as output easily applied files for the three types of program."

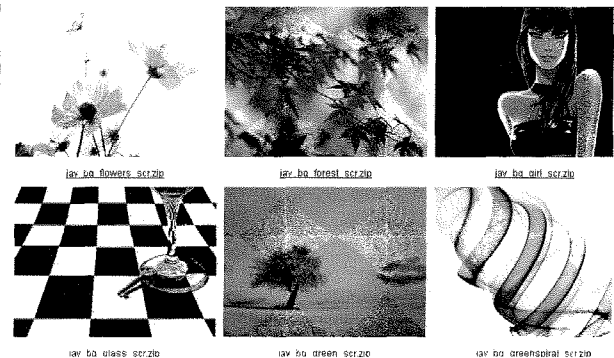
WALLPAPER

Javier Guerra writes:

"I have prepared 30 images for wallpapers of QL in 4 colours in 512x256 mode to use with the utility "wallpaper", or for other uses, under Pointer Environment. you can download them from here:

<http://badared.com/badaman/QL/images/index.htm>

I will select 22 to make a disk image of 720 Kb. All the images are under Creative Commons licence, so they can be used freely."



How Many Members?

How many members does Quanta really have? A QL Today investigation has revealed that, although Quanta has suffered a severe loss in membership in recent years, this may not be as huge as Quanta's official membership figures suggest.

Quanta's own statistics suggest during the two year watch of the present officers 28% of the members have left the organisation. QL Today believes the true figure to be nearer 22%.

Problems arose last year when the committee ended the long standing tradition of reporting the membership as at the previous 31st December. This date gives the most accurate figures as by then all subscriptions are in. At last year's AGM Quanta reported that it had 161 members on 31st January 2006. QL Today's editor, Geoff Wicks, queried this figure as it would represent a loss of 111 members or a fall of 41% of the membership in 13 months. A revised membership figure of 219 was later reported in the Quanta Magazine, but even this would seem to be a serious underestimate.

219 members would give Quanta a subscription income of £3,000 to £3,100. In practice Quanta's 2005 subscription income was £3,426. This would suggest that Quanta membership at the end of 2005 was not 161 nor even 219, but approximately 244. Similarly at this year's AGM Quanta membership was given as 196. An analysis of 2006 subscription income would indicate a membership at the end of the year of about 213.

The good news is that the drop of membership in 2005 was not 53 as the official figures suggest, but only 28. The bad news is that subscription income fell by 8.8% in 2004, 10.4% in 2005 and 12.8% in 2006. This suggests the fall in Quanta membership is not being arrested, but is slightly accelerating in both relative and absolute terms.

Gremlins ruled, OK!

The last issue of QL Today was badly infected with Gremlins.

Readers may have had a sense of déjà vu when they read "Byts of Wood". This was not because Roy Wood had nothing new to say, but because we accidentally reprinted the old text from issue 3 instead of the new text Roy had prepared for issue 4.

Our apologies to Roy and all our readers.

Two errors crept into our news report of various Quanta matters.

We reported Quanta's income had "fallen by over 18% from £4,616 to £3,400". This would have been a fall of 26%. The editor had accidentally used an income subtotal and not the grand total. The sentence should read "fallen by over 18% from £4,616 to £3,748".

We also reported "Workshop costs have risen from £395 to £1,047" and in so doing repeated a lack of clarity that had also appeared in the Quanta financial report. The show costs in 2006 were at an average level, but those in 2005 were abnormally low.

Finally there was a nasty outbreak of green-grocers' apostrophes in our humorous piece "PCs going cheap". Should you see the editor standing in a corner with a dunce's cap on his head at the next QL show, you will now know why.

Electronic QL Today

by Geoff Wicks

QL Today has received several suggestions to produce an electronic version. Simon Balderson was the first. He writes:

Are there any plans to follow QUANTA's lead and publish an on-line version of QL Today? It would help save on mailing costs and I for one would be interested in subscribing to an electronic version. I know there were concerns about file size of individual issues of QUANTA magazine and people who did not have broadband access being unable to download them. I don't see a file size of a few megabytes as too much of a problem as I currently subscribe to Everyday Practical Electronics in electronic form which has a file size of around 10Mb and I only have dial-up access.

I do like to keep my back issues of QL Today but they do take up rather a lot of space in paper form. Now that QUANTA is in electronic form I might even re-subscribe to it! Seeing as QL emulators are probably the most popular platform for running QDOS/SMSQ/E viewing the magazine with Adobe Acrobat reader shouldn't be much of a problem.

Mr. Balderson raises a question that other QL Today readers are asking. We have to disappoint him with a firm "no" for various practical and technical reasons some of which we had already written about in vol. 11 issue 2.

Even in the most favourable circumstances the file size of an electronic QL Today would be several megabytes, because the average issue of QL Today has about three and a half times the content of the average issue of Quanta Magazine.

The Quanta committee have set a maximum file size of 500Kb for the electronic magazine, which is arguably too low. To achieve the low file size they have had to place restrictions on the content of the magazine, the most obvious of which is the removal of all display advertising. They may also have to restrict the number of illustrations in individual issues of the magazine. To make up the net loss of advertising revenue they will need a minimum of 40 members opting for the electronic version of the magazine.

QL Today would not wish to have similar restrictions as we feel it would seriously compromise the editorial content of the magazine. This means that even in the most favourable circumstances the file size of an electronic QL Today would probably be over 5 times that of the Quanta Magazine.

There is a further serious technical complication, which means that we are far from having ideal circumstances. QL Today is produced with Calamus, which is an emulated ATARI program that does not have PC fonts. This means that a PDF file has to be written pixel by pixel as a bit map. This takes many hours to do and produces a huge file size. Jochen's tests showed that a PDF file in greyscale took over 24 hours to generate and would not fit on a CD. It had to go onto DVD. Issues 4 and 5 of volume 10 were produced as an experiment as PDF files, but the quality was disappointing. We reverted to the old production techniques for volume 11.

One of our readers suggested not to use Calamus but a different PC DTP program. Well, the quality of QL Today could not be maintained with other programs, simply, because they are either not as good as Calamus or, if complex, expensive. Such programs inevitably require a steep learning curve and it would take Jochen several months to get good results out of them. He cannot dedicate 100% of his time to QL Today production. Jochen tried a different DTP program some time ago, but soon stopped as it was clear that the result would not reach the current QL Today quality, layout-wise. Even then, considering the amount of information in QL Today, a 60-page issue with pictures would still not fit into 10 or 20MB. How many readers would download 50 or 100MB?

There are other considerations. Postage costs, especially to our overseas readers, make up the largest part of the QL Today price. The price can only be what it is now, because Jochen does NOT post the issues from Germany. They are mailed from Holland (which costs petrol to get there) or shipped to the UK (higher UPS shipping costs). If they were being sent from Germany, the postage would go up by 60%, compared to the cost of sending them from Holland.

There is another difference between ourselves and Quanta. All our readers would have to opt for an electronic magazine, because we cannot publish QL Today in two ways. This is because we ship the magazine in bulk between Germany and the UK. Every lost paper-copy would result in higher per-issue shipping costs. We have already reached a level where every lost user adds considerably to the cost of shipping the magazines to Roy. It did not make much difference when 1000 issues were shipped, and it dropped to 900. It would make a difference, if the number were to drop from 100 to 50, because 50 readers decide to go for an electronic version (the shipping costs do not simply halve just because the weight halves!). It is also not worthwhile for Jochen to drive to Holland to mail four US QL Today issues instead of 10.

We have already passed the point where things automatically become more expensive: business mail in Germany needs at least 50 letters with a destination inside Germany, to get the best rates. Well, we have been under 50 now for 2 years.

Even if we produced an electronic version of QL Today of say 50MB, what about the readers who do not have DSL or flat rate? If even as many as 80% of our readers were able to download it, what do we do about the other 20%?

We are stuck with the fact that with the current system, it is technically impossible anyway. But even if we find a way round this, how would we deal with readers who depend on paper copies? We have a similar problem to Quanta, but because we produce many more pages in A4 and therefore much heavier material, the postage plays a much larger role.

I guess that the day the postage reaches an unacceptable limit (and it is already quite high), or the day we lose more readers, it will be the end of QL Today. Let us hope that it will not happen soon.

[Jochen's comment: I will look into the possibility of producing a cover-DVD containing back-issues of QL Today at the end of Volume 12. No promise, but I'll try to save your storage space.]

QUANTA



Independent QL Users Group

World-wide Membership is by subscription only,
offering the following benefits:

Bimonthly Newsletter – up to 40 pages

Massive Software Library – All Free!

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: MasterCard: Accepted

Now in our Twenty Fourth Year

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane
Stretford, Manchester, M32 9EH (UK).**

Tel. +44 (0) 161 865 2872

Or

Visit the New Quanta Web Site

<http://www.quanta.org.uk>

E-mail: membership@quanta.org.uk

Archivers Cover Disk

by Dilwyn Jones

As part of the follow-up to the "Zip And Unzip" article in the last issue, we thought we'd bring you a really useful cover disk which includes not only the Zip and Unzip programs, but most of the freely available archiving programs for QL and compatible systems.

And in addition to the archiving programs, we thought we'd include Thierry Godefroy's wonderful Archivers Control Panel as well. This program is a user-friendly pointer driven front end program for 6 of the most commonly available archiving applications - Arc, LHA, LHQ, Zip, Zoo and Tar.

Archivers Control Panel isn't an archiving and compression program in itself. It simply acts as a controlling program for the other packages listed above, making them easier to use by replacing the need to remember and work out often complex command lines needed to drive the archiving programs themselves. You will remember from the first article how complex, tedious, repetitive and sometimes downright unfriendly these command lines can be!

As there was a little bit of space available on the disk, we decided to include as many QL archivers as we could fit on. Those included on the disk are:

1. Zip and Unzip

These are the most common archiver programs for the QL. These are the Jonathan Hudson ports of the official Infozip Group programs.

INSTRUCTIONS: IZREADME_SMS and UNZIP_DOC (both plain text files)

PROGRAMS: ZIP and UNZIP

2. LHA and LHQ

Franz Herrmann's port of these programs.

INSTRUCTIONS: LHA.TXT (plain text). No specific instructions for LHQ.

PROGRAMS: LHA and LHQ

3. ARC

Jeremy Allison's QL port of the ARC program.

INSTRUCTIONS: ARC_DOC (Quill DOC file)

PROGRAMS: ARC

3. BZIP and BUNZIP

Thierry Godefroy's port of the BZIP program, which typically manages 10 to 15% better compression than Zip, but generally needs much

more memory than the Zip system.

INSTRUCTIONS: BZIP2_QDOS.TXT, BZIP2_README and BZIP2.TXT (all plain text files)

PROGRAMS: BZIP2 and BUNZIP2

4. GZIP

Thierry Godefroy's port of the GZIP program for QDOS.

INSTRUCTIONS: GZIP_QDOS.TXT, GZIP.TXT and GZIP_README.TXT (all plain text files)

PROGRAMS: GZIP

5. TAR

Jonathan Hudson's port of the TAR (Tape ARchive) program. An archive is a single file into which several files are packed. These files are not necessarily compressed, but TAR can work with GZIP or the Compress program on the disk to handle compressed files.

INSTRUCTIONS: TAR_MAN, TAR_README and TAR_README_SMS (all plain text files)

PROGRAMS: TAR and COMPRESS

6. UNARJ

QL port, by Derek Stewart, of the UNRAR utility for decoding ARJ files.

INSTRUCTIONS: UNARJ.TXT (plain text file)

PROGRAMS: UNARJ_EXE

7. UNRAR

Derek Stewart's port of the UNRAR program, which decodes archives created with the RAR program.

INSTRUCTIONS: UNRAR_QDOS.TXT and UNRAR_LICENSE.TXT (plain text)

PROGRAMS: UNRAR_EXE

8. ZOO

A QL port of the Zoo archiver, by Franz Herrmann.

INSTRUCTIONS: ZOO_README and ZOO.TXT (plain text).

PROGRAMS: ZOO

Due to space limitations on the cover disk, I've only placed the essential files here. You can get full versions of most of these packages on the websites of the authors, or from the Archivers page on my website:

<http://www.dilwyn.uk6.net/arch/index.html>

BOOT Program

Some of these programs require various extension files, such as Richard Zidlicky's Signal Extension (SIGEXT30_REXT), the Environment Variables extensions (ENV_BIN) from the C68 package, Pointer Environment (PTR_GEN, WMAN, HOT_REXT), Thierry Godefroy's modified QLiberator Runtime extensions (QLIB_RUN336MOD) and the Jochen Merz menu extensions (MENU_REXT). This latter extensions file is not supplied as it is copyrighted commercial software – add it to the disk if you have a copy). MENU_REXT is not essential, most programs will work to some extent without it, although you won't have the convenience of the user friendly menus for example.

Figure 1 shows a listing of the BOOT program on the disk, which installs extensions and starts the Archivers Control Panel program ACP_OBJ.

Figure 1 - Boot Program

```
100 REMark QL Today Archivers cover disk boot program
110 REMark by Dilwyn Jones 2006
120 :
130 TK2_EXT : REMark activate Toolkit 2 if required
140 :
150 drive$ = 'FLP1_' : REMark extensions loaded from here
160 :
170 version$ = VER$
180 IF NOT (version$ == 'HBA') THEN
190   REMark install pointer environment v2.01 unless on SMSQ/E
200   LRESPR drive$ & 'ptr_gen' : REMark pointer interface
210   LRESPR drive$ & 'wman'      : REMark window manager
220   LRESPR drive$ & 'hot_rext' : REMark hotkey system II
230 END IF
240 :
250 REMark Menu Extension
260 LRESPR drive$ & 'menu_rext'
270 :
280 REMark Environment variables
290 LRESPR drive$ & 'env_bin'
300 :
310 REMark Signal Extensions
320 LRESPR drive$ & 'sigext30_rext'
330 :
340 REMark modified QLiberator runtimes
350 LRESPR drive$ & 'QLib_run336mod'
360 :
370 REMark activate hotkeys
380 HOT_GO
390 :
400 DATA_USE drive$ : REMark DATA_USE default drive
410 PROG_USE drive$ : REMark PROG_USE default drive
420 :
430 EX drive$ & 'acp_obj'
```

Line 130 activates Toolkit 2. Remove this line if a TK2_EXT command is not needed to activate Toolkit 2 on your system, e.g. if you have the Care/QJump EPROM version. Some disk interfaces may need a TK2_EXT command to 'wake up' Toolkit 2 if they do not activate it automatically at switch on.

Line 150 sets the variable 'drive\$' to the name of the drive from which the files are loaded. Alter this as required for your system.

Line 170 checks the version of BASIC on your system. SMSQ/E includes a version of pointer environment equivalent to the PTR_GEN, HOT_REXT and WMAN loaded in lines 200 to

220, so the test in line 180 seeks to check if the QDOS pointer environment files PTR_GEN, WMAN and HOT_REXT need to be installed. This will not work on a QXL running the original SMSQ which does not have pointer environment built in, so if using the disk on a QXL with vanilla SMSQ, remove the IF clause in lines 180 and 230.

Line 260 installs the menu extension file MENU_REXT. If you do not have a copy of MENU_REXT, simply remove line 260.

Line 290 installs the Environment Variables extension.

Line 320 installs the Signal Extensions file.

Line 350 installs a modified version of the QLiberator compiler runtime extensions, called QLIB_RUN336MOD. This is a patched version 3.36 of these extensions, altered to correct a small bug in the original QLIB_RUN, which means that error number and error line numbers can get reversed. Some of Thierry Godefroy's programs rely on having this version installed.

Line 380 issues a HOT_GO command to ensure that the hotkey system is activated. Without a HOT_GO command, hotkeys might not work. Please note that if you modify this boot program to load other extensions, you must install them before issuing the HOT_GO command, since HOT_GO starts a job running and causes RESPR and LRESPR commands not to work properly. Line 400 and 410 sets PROG_USE and DATA_USE settings to the drive name where the programs are stored. Some of these programs

(including Archivers Control Panel) look at these settings, so it is important that they are set to where the programs reside.

Line 430 starts the Archivers Control Panel program.

Configuration

Some of the programs (e.g. ACP_OBJ) can have their default settings altered by means of the standard CONFIG program. See individual program instructions for details of what's possible. By all means experiment, but make sure you have made a backup copy of the cover disk which can be used for recovery if something goes horribly wrong and files get damaged!

Archivers Control Panel

This is a pretty complex program able to handle up to six different types of archiver program. It has an initially bewildering number of options and actions which can be used, but once you get used to it, it quickly becomes quick and convenient to use. This program is the subject of a complete separate article in this magazine.

Please note that we've only put the English language version of Archivers Control Panel program on this disk. French speakers can obtain a French language version from Thierry Godefroy's website:

<http://qdos.dyns.net/>

or

<http://thgodef.nerim.net/smsq/>

Review of QL Wiki

by Gerard Phelan

When Rich Mellor announced to the QL web community that he had created a QL Wiki, the initial reaction was puzzlement and then annoyance that the existing information at http://en.wikipedia.org/wiki/Sinclair_QL was considered inadequate and was to be duplicated. Some people still hold those views, but search on the big Wiki for QL and you will also be offered the Phillips QL and the Bedford QL. Search for Gold Card and you are offered a Credit card or a UK rail season ticket. Rich's idea is that that the Wikipedia entries will provide the global introduction, but the QL Wiki will provide a closed world in which you do not have to continually choose to see things the QL way and where a structure can be provided to help the user ex-

plore the wide world through QL eyes.

Beginning at http://www.rwapadventures.com/ql_wiki/ the Wiki front page is adorned with a traditional QL picture. A brief introduction to the QL is followed by the first of the Content hierarchies:

History – How the Sinclair QL has changed over the years

Platforms – The various computers and emulators, which can run QL software

Hardware – The expansion cards, which have been created over the years

Operating Systems – A guide to the different versions

Software – The numerous software titles

Traders and Personalities – Who's who within the QL community

CD-ROMS – A wealth of information and programs

Links – Links to other web sites related to the Sinclair QL

Everything can be edited if you wish, by clicking on an edit button. Thus by the time you read this, the content list might be quite different. On the left hand side there's another contents list, which is not quite the same, being system generated in alphabetic order. This list also includes information on Wiki editing and access to a Wiki Sandbox for testing.

A Wiki is a reference guide created by the readership at large. By definition a new Wiki does not have very much content. It would therefore be silly to comment much on what entries are present today so I have focussed my comments towards the current structure. Even this can be changed by anyone – with a little work.

History

Today this is a straightforward article based on the Wikipedia entry.

Platforms

This is organised as a comprehensive list for Q40/60, QLAY, QPC, uQLx, each of which open up in their own sections. There are already 12 platforms listed. This alone demonstrates the benefit of the closed QL Wiki design because here they are grouped and await expansion of their content by the knowledgeable.

Hardware

Already you will find over 90 entries, grouped in alphabetic sections, most/all with descriptions by Rich or Dilwyn Jones. There is plenty of scope for increasing the detail on each entry.

Operating Systems

Unlike the Hardware section, today this is a single page with brief descriptions of QDOS, Argos, SMS2, and all the rest. Were someone keen to explain the architecture of SMSQ/E then they could create a hierarchy to provide the room needed.

Software

Rich writes this is intended as a *"Full list of the Software titles published for the Sinclair QL"*. Over the years he has dealt with so much software that his list of over 60 titles is certainly a good start. Wisely he has included sub groups,

initially Games, Utilities and Front Ends, to cover the likely expansion as the enormous list of software is added.

Traders and Personalities

Rich has a long association with the QL so there are already over 40 traders listed, although few today have content. The system-generated menu on the left shows just the pages that have content, which is helpful. Today there are just five QL personalities. What a self-effacing community the QL enthusiasts are!

CD-ROMS

Today this is already a comprehensive looking treatise on the technical aspects of QL CD-ROMS and a listing of what you can get if you can master the subject. One advantage of the Wiki format will be that others will be able to restructure this into a really non-technical introduction and immediately get into listing the goodies, leaving the technical stuff for those who want to read it. Readers of Wikipedia will see this type of desirable change is flagged up all the time by the self appointed administrators.

Links

It's just a list – however being a Wiki, this means that dead links can be removed and new ones added without having to write emails or ask Rich to do something.

Cool toys to play with

On the left below the menu, there are further options

Search – Text search box. Seems to work quickly.

Metadata – Not sure yet what this does for us

Recent changes – This is system generated and does what it says

What links here – Mostly of use to those making changes

Site plan – This provides a great overview

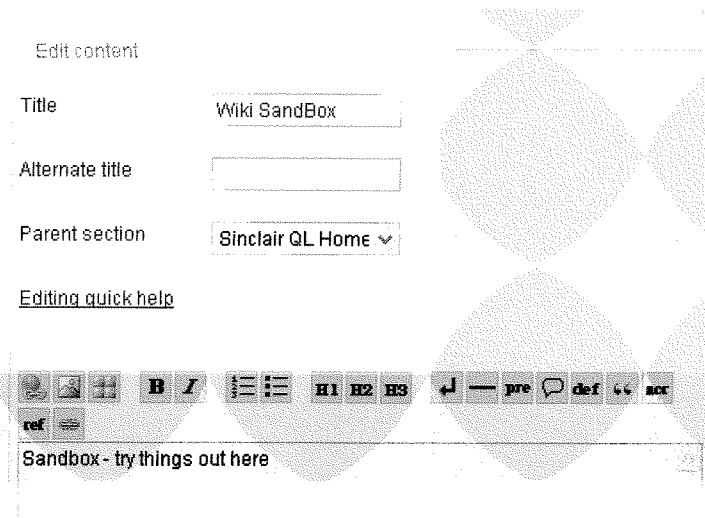
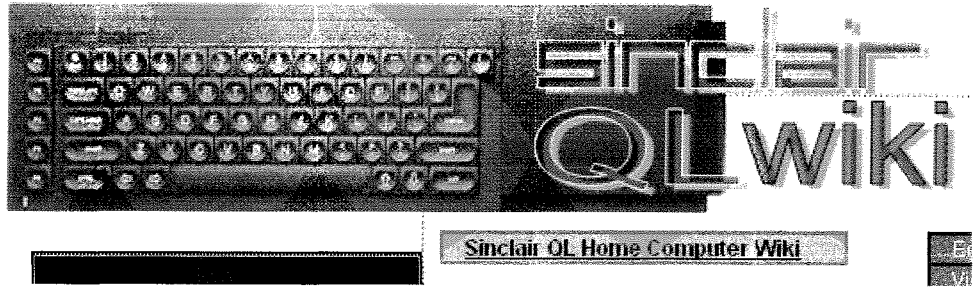
Various RSS links – RSS is a system for providing news feeds about a system. I was unable to see if this was configured, since I am not setup to use an RSS facility.

Updating the Wiki

"OK this is not always as easy as it looks" is how Rich introduces the documentation, and a brief read confirms this. *"First of all you have to register as a user and then login to the site be-*

fore you can edit a page". Clicking on one of the edit buttons opens an edit window with 20 editing icons, all with popup descriptions when you hover the mouse on them. You can use a variety of simple HTML and you can probably embed or link to images, but that part of the Help description awaits completion as I write this review. For making simple changes, you do not need worry about any of this. I changed the QPC2 description by changing "I" to "Marcel" in less time than taken to write here that I did so. This type of end user update can rapidly enhance the quality of a Wiki by bringing together the knowledge and proof-reading skills of the whole community. In this instance "I" is incorrect in a Wiki text, because although Marcel Kilgus wrote the original description, it ceases to belong to anyone in particular once added to a Wiki and could be updated next day by a complete stranger. Creating page links will require some planning and careful execution, but that would be the case in any hyperlinked environment.

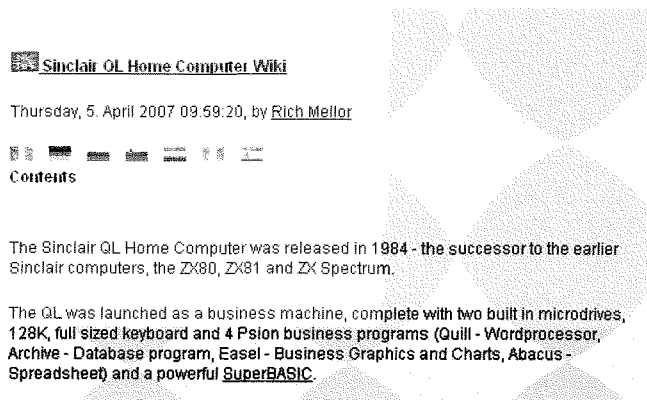
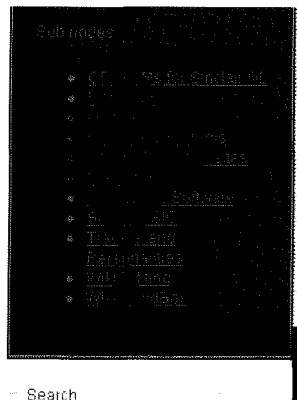
Design and appearance of the QL Wiki



Ah! The best part is that Rich has said the design is based on CSS style sheets, which he is willing to let someone knowledgeable update. Underlying the top part of each page is a pale image of the original QL screen, its pale red and grey blocks add nothing to the beauty and take away some of the clarity. The system-generated menu on the left hand is in a small red font on a black background. It would be possible to make it less readable, but not by very much! At the top of each page is a title below which is a row of flags. Clicking on the relevant flag displays the contents in English, German, Russian, Slovene, Dutch, Italian or Spanish. Great in principle, but many of them have no content and immediately launch the page editor, which is an optimistic way to recruit translators! Below the flags there may be a "Contents" subheading. Just hovering the mouse pointer over this pops up a list of the sub-headings on the page - if the author has used the right codes when setting them up. This seems more quirky than worthwhile. On the CD-ROMS page the sub-heading list fills half the page and as you move the mouse pointer the start of the text continually bobs up and down. In my book this is a possible maybe - for deletion. It is not at all obvious that the way you get back to the start is to click on the black text in the

Speed

Using a 1Mb broadband link the navigation from page to page is languid, but not slow. Opening a page for editing can take 30 seconds but is often quicker.



green box at the top reading "Sinclair QL Home Computer Wiki". The web world has generally agreed on "Home" as the text meaning go to the beginning, and it's too late in the day to propose "SQHCW" as a new paradigm! This is a more important issue than it may appear at first because if you linger too long on a page then attempting to return using your browser BACK button may give you the message that the previous Web page has expired.

Finally the left hand 25% of the page, below the system generated menu and other links is empty. The same happens in Wikipedia, but there the unused left border is 15% of the page, which leave much more for the text you want to read

Summary

In QL Wiki I think Rich has provided the basis of a most useful tool. There have been attempts before to provide the QL community with reference information. Those attempts have been limited by the need to focus updates on a single editor who inevitably has the rest of life to experience as well as that of the QL and is hence self limiting and in the end it stops working. The Wiki model removes that restriction. It will need all of us to do our part, but I'm looking forward through rooting around in my QL library for obscured but still valuable information worth sharing. If Rich starts to complain that his disc space limit has been reached, we will know the venture has been a success!

Working Space

by George Gwilt

The manipulation of working space in an assembly program can take many forms and, in the hope that it may be considered useful, I describe some of my methods. These methods were chosen with two principles in mind.

First, the space used should be as small as possible.

Second, the methods of using heap space should minimise heap fragmentation. Since one way of achieving this aim is to allocate large chunks from the heap rather than small this conflicts with the first principle so interesting compromises must be made.

I am going to consider the following three areas for the storage of information:

- The program itself.
- The program's data space (which includes the stack).
- The free memory or heap.

It is of course also possible to store temporary information in files placed on hard drives or floppies but I am restricting myself here to the three areas listed above.

The Program Itself

When I first started writing assembler programs I placed intermediate results in space allocated inside the program itself. For example if I wanted to store an integer representing, say, a number of choices I would put in the program:

```
NCH      DS.W      1          ; Reserve a space of 1 word
```

If the number I wanted to store was in D4.W I would store this by

```
LEA      NCH,A0      ; Set the address of NCH in A0
MOVE.W   D4,(A0)     ; Move the contents of D4.W to NCH
```

This is perfectly reasonable. It works well but has two drawbacks. The first of these is that the program is now not re-entrant and the second is that you need two instructions for the storing of D4.W. In many cases it does not matter one bit whether the program is or is not re-entrant. So why the implied criticism?

The answer to that is that if you want to have, say, three of your non re-entrant programs running at the same time you will need to have loaded in program space three versions of the program each including the code as well as the data space.

However, if the program does not alter itself, it is thereby "re-entrant" and only one version of the code need be loaded. It is only the dataspace which need be set up for each of the programs running. Each of these programs could simultaneously have program counters pointing to code in different places. It might seem difficult to have several versions of a program running all using the same program code. In fact it is quite easy using the hotkey system.

Typing `ERT HOT_RES('t','win1_exe_prog')` followed by successively pressing `ALT t` will cause several versions of "prog" to be started.

The second drawback, admittedly minor, was that two instructions were needed to store the information in the program space. In the following section I show how to use the data space and so allow the program to be re-entrant.

The Program's Data Space

When a program is started, for example by EX, all the registers are set to zero except for A4 to A7. A6 points to the start of code and (A6,A4.L) points to the start of data space. I find it useful in my programs to have A6 pointing to the data space. I do this by:

```
LEA      (A6,A4.L),A6      ; A6 -> start of data space
```

This allows me easily to store and retrieve information from data space provided I know the position of the item. For example, if the number I want is 42 bytes from the start of data space storing the item needs just one instruction:

```
MOVE.W   D4,42(A6)
```

If you use an assembler with an RS facility it is easy to define all the offsets needed. For example:

```
RSSET    0          ; Sets the RS position at 0
EQB_LST  RS.L      1          ; EQUates EQB_LST to 0 and advances RS to 4
EQB_LEN  RS.L      1          ; EQUates EQB_LEN to 4 and advances RS to 8
. . . .
```

Individual Items

The items that can economically be stored in data space are those of fixed length. To store a list of an indeterminate number of items is more wasteful, but it can be done if there is a defined maximum number.

Indeterminate Number of Items

An example of this is general working space. I would set a 256 byte buffer in data space into which to read lines of an ASCII file to be processed if I knew that each line would not be contain more than 256 characters.

Here space must be made available for the maximum number. If there are several groups of items each of indeterminate number the data space may become prohibitively large. It is for this reason that I use also free memory for storage of items and for working space.

```
*****
*                                          *
*   Data Space Start -> |                *
*                       |   Data       *
*                       |                *
*                       |   .          *
*                       |                *
*                       |   .          *
*                       |                *
*   Data Space End   -> |   Stack      *
*                       |                *
*                       |                *
*****
```

Before I describe how I do that I would like to mention one other aspect of the data space. The end of data space contains the stack. Data space is filled from the top and the stack from the bottom. It is important that there be always enough room for the stack otherwise some of the items towards the end of the data space could get overwritten. The stack will be used by subroutines for storing registers and intermediate results. It will usually be the case that the maximum amount of stack needed for any subroutine is known and this information should be taken into account when the size of data space is originally set by the programmer. However, problems arise when subroutines can call themselves an indeterminate number of times. In such cases, instead of imposing a maximum number of recursions, I set a check at the start of such routines that there is currently enough stack space left. If there is not enough space the program can be stopped with the message that data space should be increased.

The Heap

A User Heap

The remaining problems of storage all relate to indeterminate numbers and indeterminate sizes. This seemed to me a severe problem with my assembler program GWASS. Here there could be a small, or large number of labels or of included files or of IF clauses all this depending on the particular programs GWASS was required to assemble. I thought it better not to set a maximum number for each category, especially since it was likely that there would usually be a mixture of small and large numbers in the various categories so that the overall space needed would be less than that needed to allow for each maximum. I decided to do two things. First I would use increasable storage. By that I mean that I would allocate relatively small amounts of storage for each set of items and increase the amount only when needed. Second I would apply that principle to the heap itself.

Thus I allocate a decent sized chunk of the common heap at the outset. This is managed as a user heap for the storage of items. If the user heap space becomes filled I grab further chunks from the common heap and add them to the user space, keeping a note of their addresses.

When an assembly is complete any of the additional chunks allocated from the common heap are returned and the user heap is reset.

Let's see how the user space is set up in the first place. This is done in two stages. First we must get hold of space from the common heap for ourselves by:

```

MOVE.L    #HEAP_SPACE,A1      ; The amount of space needed
MOVEQ    #-1,D2                ; Space marked for this job
MOVEQ    #MT_ALCHP,D0         ; Grab the . .
TRAP     #1                    ; . . space
TST.L    D0                    ; OK? . .
BNE      OOPS                  ; . . NO!

```

The address of the allocated space is now in A0 and the number of bytes allocated is in D1.L

The next step is to define this space as a user heap by using MM_LNKFR:

```

LEA      HEAPAD(A6),A1        ; A1 contains the pointer to the pointer
                                           ; to the existing user space
CLR.L    (A1)                  ; see that this is zero to indicate
                                           ; no user space yet
MOVEA.W  MM_LNKFR,A2         ; Do the linking
JSR      (A2)

```

Now the contents of HEAPAD(A6) constitute a pointer to the user heap whose length is HEAP_SPACE.

Space can now be allocated from this user heap by using MM_ALLOC. To produce, for example, a 32 byte space we do:

```

MOVEQ    #32,D1                ; Number of bytes needed
LEA      HEAPAD(A6),A0         ; Ptr to ptr to space in A0

```


Kaiser-Wilh.-Str. 302 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
<http://smsq.j-m-s.com> smsq@j-m-s.com

QDT - The QL DeskTop

QDT brings the QL a modern, easy- and intuitive-to-use graphical user interface. Arrange your programs, games, applications in folders, start your favourite applications with a single click. Fully configurable!

QDT comes with an automatic installer – like a guided tour. Installing it is as easy as never before!

System requirements:

- SMSQ/E Version 3.06 or later
- Graphics Driver 2 Colour support
- HARD DRIVE with at least 3MB free space.
- 4MB RAM

What do you get when you order QDT?

- A floppy disk for Aurora or quick-start users.
- A CD ROM containing a QXL.win file, backgrounds, sprite editor, and many other extras.
- A printed quick-start guide

QDT costs EUR 59,90 plus postage (EUR 3,- Europe, EUR 6,- rest of the world).

BALLED! - game for the Pointer Environment

Balled is the new Game from Wolfgang Lernerz for QDOS and SMSQ/E. It runs in high-colour mode (looks great) and can also be played in standard QL Mode 4.

Also, Balled uses the digital sound system if you run it on a recent version of SMSQ/E and QPC, Q40 and Q60! The aim of the game is to build lines by moving coloured balls around, following some rules, of course. After every move, more balls appear, so you have to be careful not to fill up the board – then the game is over! Different levels of difficulty and wildcard balls are also featured. Find out how to get bonus and higher scores to fill up the high-score table!

Balled costs only EUR 11,90 plus postage (EUR 2,- Europe, EUR 4,- rest of the world).

QPCPrint - printer driver for QPC

QPCPrint allows you to print to virtually any printer connected to your PC running QPC, even to fax and pdf printer drivers. It accepts EPSON ESC/P2 codes to any PAR printer and converts it to output which can be handled by Windows, which looks very similar to original EPSON output. You can even configure the individual fonts used by the emulation. This is THE application many QPC users waited for a long time!

QPCPrint costs only EUR 39,90 plus postage (EUR 3,- worldwide - shipped on CD)

Free Delivery via email now possible: <http://Printer-Emulator.J-M-S.com>

ZX8301 only EUR 4,- each (plus 2,50 EUR postage for up to 5 ICs)

ZX8302 only EUR 2,- each (plus 2,50 EUR postage for up to 5 ICs)

Still searching for **DD Disks**? Well, I've got them as well. New, in boxes of 10 ... various colours or just blue: Pack of 10 DD disks for only EUR 4,- each (plus 4,50 EUR postage for 1 or 2 packs, 9 EUR for up to 7 packs)

If you wish to place your order via internet, please do not send your credit card details in an email!

Please use the secure contact form on the SMSQ homepage: SMSQ.J-M-S.COM

We accept VISA, MasterCard and Diners Club!

L1 EQU L2 + L3 - L4 + L5 . .

where at least some of L2 to L5 are themselves as yet undefined. To deal with such cases I set label L1 pointing to a list of op/label pairs. Each op was either + or -. Such a list I call an EQU list. In this case the list contains

+ L2, + L3, - L4, + L5 etc

To see how complicated this gets you can suppose some of L2 to L5 are themselves defined in terms of yet other labels. Incidentally GWASS draws the line when the definition of a label includes itself in the unravelled list. This is just not allowed.

Dynamic storage was used to set a space in which to make up the EQU list during the routine called GNUM, for Get NUMber, which values a label. At the routine's exit the EQU list is written out to new space allocated for it then. The space at the dynamic storage pointer EQB_LST was used to produce the particular list during one run of GNUM. The space at EQB_LST thus only needed to be expanded when a new EQU list proved too large for the existing size of EQB.

When GWASS was upgraded to evaluate "expressions" it became necessary to allow for multiple EQU lists. That was because in evaluating such an expression as

L1*(L3+L4*L5)

for example, GNUM calls itself every time it encounters either a left parenthesis "(" or the imminent arrival of an operator of higher priority than the last as with L4*L5. Each time GNUM is re-entered it may have to initiate another EQU list. Clearly one dynamic storage area will not do. But how deal with an indeterminate number of such areas?

I decided to redefine the dynamic storage area EQB as follows.

```
*****
* The new format of EQB is:
*
*   W       L       L           L
*   |-----|
*   | k | a1 | a2 | . . . | ak |
*
* k is the number of addresses a1, a2 etc
* Each address points to a 32-byte area which can be used by GNUM. When
* an address is allocated its top bit is set. This is cleared when the
* area is no longer needed by GNUM.
*
* The initial value of k is 0. If an address is needed, EQB is extended
* by a routine GSP. Each extension adds 15 more addresses to the list.
*
* Each extension sets up a 480-byte area divided into 15 32-byte
* sub-areas whose addresses are held in EQB.
*
* The sub-areas so formed will contain all the "op"/"label" pairs needed
* in the final EQU list in the format:
*
*   L       L       L
*   |-----|
*   | t | pr1 | pr2 | . . . | prn |
*
* t is either the number of "op"/"label" pairs (pr1, pr2 etc) or
* the pointer to a further chunk containing more pairs. If t < 8 it is
* the number of pairs in the sub-area. Otherwise it is a pointer to the
* next block.
```

```

*
* Two routines operate this system:
* GN_GTL returns the address of a sub-area.
* GN_RTA resets an area to "available".
*
* The first long word at the address returned by GN_GTL has been zeroed
* to show it empty.
*****

```

This system enables an indefinite number of EQU lists to be generated during a call to GNUM.

There may indeed be, and probably are, simpler ways of dealing with this particular problem. However, there is always a balance between taking a long time to find the theoretically ideal coding and actually finding fairly quickly a method that works.

Now that I have exposed my method, perhaps Norman Dunbar will show us a better one!

The Coding

I give here the coding for several routines mentioned above. The routines are:

Name	Purpose
GSP	Get space for dynamic storage
EXTENS	Extend dynamic storage
NN_LG2	Get space from a possibly extended user heap
ALC	Allocate space from user heap
REL_SPC	Return additional space to the common heap
GN_RTA	Mark address in A2 as available for an EQU list
GN_GTL	Return an available address in A0 for an EQU list

*** GSP ***

The above definition of EQB referred to three subroutines, GSP, GN_GTL and GN_RTA. The most important of these is GSP. This is a general routine for all dynamic storage. Its job is to see that the area pointed to, by the contents of EQB_LST(A6) in the above example, contains enough space. If not GSP arranges to allocate more space from the user heap and then updates the pointer. Some types of dynamic storage contain absolute addresses pointing to parts of the storage. Obviously these have to be updated when an area is extended. Also it can happen that an area has to be filled with -1 before use. Such relocation or filling of space is signed to GSP by one of the codes -1 to 2.

Thus to arrange that CH_LST points to a space containing 16 bytes we do:

```

PEA    CH_LST(A6)    ; Store the address CH_LST(A6)
PEA    16             ; Number of bytes needed
PEA    0              ; No relocation
BSR    GSP

```

We can now be sure that CH_LST(A6) points to a space containing at least 16 bytes.

Coding for the subroutine GSP:

```

;
; At entry 0(A7) = return address for GSP
;      4(A7) = 1,2 for relocation of new area = 0,-1 otherwise
;      8(A7) = space needed
;      $C(A7) = pointer to pointer to area
;
GSP    MOVEM.L  D0/A1,-(A7)
        MOVE.L  $10(A7),D0          ; space needed
        MOVEA.L $14(A7),A1         ; pointer to area
        TST.L   (A1)                ; Any space yet? . .

```

```

        BEQ      GSP2          ; . . no
        CMP.L   4(A1),D0      ; enough space? . .
        BLE     GSP1          ; . . yes
GSP2    BSR     EXTENS        ; . . no - get it
GSP1    MOVEM.L (A7)+,D0/A1
        RTS

```

EXTENS is the routine which secures the space needed for the expanded storage. The contents of the original block are copied to the new space and relocation is done if needed. Finally the old space is returned to the user heap.

The allocation of new space from the user heap is carried out by the subroutine NN_LG2.

*** EXTENS ***

```

;
; EXTENS extends the length of CH_LST etc
; On entry A1 -> the list pointer, the current length and the increment
;          D0.L = space needed
;          A6 -> DATA SPACE
;          $10(A7) < 0 no relocation : space -> -1
;                  = 0 no relocation
;                  = 1 relocation @ 1st longword of space
;                  = 2 relocation @ each 6*k+2 longword of _IX space
;
; On exit  Pointer and current length are updated. If needed the old list
;          is copied to the new place and space returned to the user heap.
;          Also space is set to -1 if wanted.
; No registers are used.
;
EXTENS  MOVEM.L  D0-3/A0-3,-(A7)
        MOVE.L  4(A1),D2          ; old length
        MOVE.L  8(A1),D1          ; increment-1
        ADD.L   D1,D0             ; round to . .
        NOT.L   D1                ; . . increment
        AND.L   D0,D1             ; new length needed
        BSR     NN_LG2            ; get address of new space to A0
        BNE     QER4              ----> ERROR
        MOVEA.L (A1),A2           ; address of current space . .
        MOVEA.L A2,A3             ; . . kept in A3
        MOVE.L  A0,(A1)           ; set new address in DATA SPACE
        MOVE.L  D1,4(A1)          ; set new length
        MOVE.L  $30(A7),D0        ; relocation code
        BPL     EXTENS6           ; don't set to -1
        LSR.W   #2,D1             ; total new length/4
        BRA     EXTENS7
EXTENS8 MOVE.L   #-1,(A0)+        ; set -1
EXTENS7 DBF     D1,EXTENS8        ; count long words
        MOVEA.L (A1),A0           ; reset A0
EXTENS6 MOVE.L  D2,D1             ; old length (set for release)
        BEQ     EXTENS4           ; no old list
        MOVE.L  A0,D3             ; Find the relocation . .
        SUB.L   A2,D3             ; . . adjustment
        SUBQ.L  #1,D0             ; relocation code - 1
        BMI     EXTENS5           ; no relocation
        BNE     EXTENS9           ; relocation 2
        ADD.L   D3,(A2)           ; relocation 1
EXTENS5 LSR.L   #2,D2             ; no. of long words
        BRA     EXTENS1
EXTENS2 SWAP    D2
EXTENS3 MOVE.L  (A2)+,(A0)+       ; copy old to new
EXTENS1 DBF     D2,EXTENS3        ; count may . .
        SWAP    D2                ; exceed . .

```

```

        DBF          D2,EXTENS2          ; . . 32768
        MOVEA.L     A3,A0                ; old address
        LEA        HEAPAD(A6),A1
        MOVEA.W     MM_LNKFR,A2          ; link back space to user heap
        JSR        (A2)
EXTENS4  MOVEM.L    (A7)+,D0-3/A0-3
        RTS
;
EXTENS9  MOVE.L    XREF_IX(A6),D0        ; pointer to space
        LEA        2(D0.L),A0            ; 1st address *
        ADD.L     IX_LEN(A6),D0         ; +1
        SUBQ.L    #4,D0                  ; last address
        BRA        EXTENS10
EXTENS11 ADD.L     D3,(A0)                ; add adjustment
        ADDQ.L    #6,A0                  ; to next entry
EXTENS10 CMPA.L    DO,A0
        BLE        EXTENS11             ; more to do
        MOVEA.L   (A1),A0                ; reset A0
        BRA        EXTENS5

```

NOTE: The Effective Address 2(D0.L) is allowed as a 68020+ instruction. It is equivalent to 2(A0) with A0 having the value in D0.L. It is also equivalent to 2(A0,D0.L) with A0 containing zero. It is a way of using D0 as if it were an address register.

*** NN_LG2 ***

The subroutine NN_LG2 allocates the requested amount of space from the user heap, expanding that if needed.

```

; NN_LG2
; SR to get space from the user heap
; On entry  D1.L = bytes to ALLOC
; On exit   DO = 0 OK or <>0 error (CCR is set)
;          A0 -> new area (1st 8 bytes cleared)
; No other regs used
;
NN_LG2  MOVEM.L    D1-4/A1-5,-(A7)
        MOVE.L    D1,D4                  ; keep space required in D4
        BSR        ALC                    ; did we get the space? . .
        BNE        NN_LG3                 ; . . no!
NN_LG8  MOVEM.L    (A7)+,D1-4/A1-5
        RTS
;
; We must now add to the user heap
;
NN_LG3  MOVEQ     #MT_FREE,D0             ; Find the amount of free space
        TRAP      #1
        CMPI.L    #ALC_MIN,D1            ; Is it enough? . .
        BLT       NN_LG8                 ; . . no - too small to ALLOC
        LSR.L     #5,D1                  ; maximum to ALLOC
        CMPI.L    #ALC_MAX,D1            ; Less than ALC_MAX? . .
        BLE       NN_LG5                 ; . . yes
        MOVE.L    #ALC_MAX,D1            ; . . no, so ALLOC ALC_MAX
NN_LG5  MOVEQ     #MT_ALCHP,D0
        TRAP      #1
        TST.L     DO                      ; Did we get it? . .
        BNE       NN_LG8                 ; . . no!
        MOVEA.L   A0,A4                    ; keep note of ALCHP'd address
        LEA       HEAPAD(A6),A1           ; Link new space . .
        MOVEA.W   MM_LNKFR,A2            ; . . into . .
        JSR       (A2)                    ; . . the user heap
        MOVEQ     #8,D1                    ; get 8 bytes
        BSR       ALC

```

```

        BNE      NN_LG8      -----> ; Hmm! can't!
        LEA      X_ALLOC1(A6),A5 ; 1st link in chain of ALLOCs
        BRA      NN_LG6
NN_LG7  MOVEA.L  DO,A5
NN_LG6  MOVE.L   (A5),DO      ; next address
        BNE      NN_LG7      ; get more
        MOVE.L  A0,(A5)      ; set new address in old link
        MOVE.L  A4,4(A0)     ; set ALCHP'd address in chain
        MOVE.L  D4,D1       ; Now, at last, get . .
        BSR     ALC         ; . . space needed . .
        BRA     NN_LG8      ; . . and return

```

*** ALC ***

```

;
; ALC sets the address of D1 bytes in A0
;
ALC     MOVEA.W  MM_ALLOC,A2   ; Vector to allocate space
        LEA     HEAPAD(A6),A0 ; Set pointer to pointer to space
        JSR     (A2)          ; Get the space
        TST.L   DO           ; Was it OK?
        BNE     ALC_1        ; No
        CLR.L   (A0)         ; Clear the first . .
        CLR.L   4(A0)        ; . . two long words of space
ALC_1   RTS

```

*** REL_SPC ***

```

;
; To release space ALCHP'd for additional user heap space
;
REL_SPC MOVE.L   X_ALLOC1(A6),DO ; start of chain
        BEQ     REL_SPC1        ; nothing to do
REL_SPC2 MOVEA.L DO,A4          ; keep in A4 for next link
        MOVEA.L DO,A0          ; address of space
        MOVEQ   #MT_RECHP,DO    ; return . .
        TRAP   #1              ; . . it
        MOVE.L (A4),DO         ; a further link? . .
        BNE     REL_SPC2        ; . . yes
        CLR.L   X_ALLOC1(A6)    ; set "no extra space"
REL_SPC1 RTS

```

*** GN_RTA ***

```

; To set the EQB address A2 as available
;
; On entry A2 = address in EQB list used in EQU list
; No registers are used
;
;
GN_RTA  MOVEM.L  DO-1/A0,-(A7)
        MOVE.W  ([EQB,A6]),DO   number of addresses
        LEA     ([EQB,A6],2),A0 -> 1st address
        BRA     GN_RTA1
GN_RTA2 MOVE.L   (A0)+,D1       address to D1
        BCLR   #31,D1          make it positive
        BEQ    GN_RTA1         it was, so try the next
        CMP.L  D1,A2           is this the address? . .
        BEQ    GN_RTA3        . . yes
GN_RTA1 DBF     DO,GN_RTA2      go through all addresses
        BRA     GN_RTA4        not found!!!
GN_RTA3 MOVE.L   D1,-4(A0)      reset positive address
GN_RTA4 MOVEM.L  (A7)+,DO-1/A0
        RTS

```


*** GN_GTL ***

```

;
; GN_GTL sets the address of a 32-byte area to A0
;
; No other registers are used
;
GN_GTL    MOVEM.L    D1/A1-2,-(A7)
          TST.L      EQB(A6)           Any addresses? . .
          BNE        GN_GTL8          . . yes
          MOVEQ      #15,D1           prepare for 1st extension
          SUBA.L     A2,A2             set 0 in A2 for no addresses
          BRA        GN_GTL9
GN_GTL8   MOVE.W    ([EQB,A6]),D1     set the count of addresses . .
          MOVEA.W    D1,A2            . . and in A2
          SUBQ.W     #1,D1            (there must be at least one!)
GN_GTL2   TST.L     ([EQB,A6],D1.W*4,2) available? . .
          DBPL      D1,GN_GTL2        . . not yet
          BPL       GN_GTL3          found
          MOVEQ      #15,D1
          ADD.W      ([EQB,A6]),D1     new count
GN_GTL9   LSL.L     #2,D1
          ADDQ.L     #2,D1            new space needed
          PEA       EQB(A6)
          MOVE.L     D1,-(A7)
          PEA       0
          BSR       GSP              extend EQB as requested
          LEA       12(A7),A7         reset stack
;
; Now we must fill the space with addresses
;
          MOVEQ      #15,D1
          LSL.L     #5,D1             480 (15 32-byte areas)
          BSR       NN_LG2           A0 -> space needed
          BNE        GN_GTL7        _____,
          MOVE.L     A2,D1            old count of addresses
          MOVE.W     A2,([EQB,A6])    set old count in new space
          LEA        ([EQB,A6],D1.W*4,2),A1 A1 -> start of new space
          MOVEQ      #15,D1          to count 15 addresses
          ADD.W      D1,([EQB,A6])    set new count
          LEA        480(A0),A2       point to END of area
          BRA        GN_GTL4
GN_GTL5   LEA       -32(A2),A2       decrease pointer by 32
          MOVE.L     A2,(A1,D1.W*4)   set address in EQB
GN_GTL4   DBF       D1,GN_GTL5
          BSET      #7,(A1)           mark "used"
GN_GTL6   CLR.L     (A0)             set initial pointer to 0
GN_GTL7   MOVEM.L  (A7)+,D1/A1-2
          RTS
GN_GTL3   MOVEA.L  ([EQB,A6],D1.W*4,2),A0
          BSET      #7,([EQB,A6],D1.W*4,2)
          BRA        GN_GTL6

```

NOTE: In the code above are various useful effective addresses allowed by the 68020+ instructions.

1. [EQB,A6] is an address equal to the contents of EQB(A6)
2. In ([EQB,A6],D1.W*4,2) the address [EQB,A6] is modified by the addition of four times the contents of D1.W and then the addition of 2.

A Weekend of Waffles?

by Geoff Wicks

Two years ago I went to the Quanta AGM at Hove sporting a magnificent black eye, which caused several people to fear that Quanta relationships had deteriorated into fisticuffs. There was a more simple explanation. I had had a nasty fall outside my hotel, and had gone inside dripping blood onto the reception desk only to discover I was in the wrong hotel.

By one of those strange coincidences that sometimes occur in life, Quanta's committee chose to use just that hotel this year. I warned them that if the hotel discovered their connection with the miscreant of two years previously, they could find themselves evicted from their warm bedrooms and forced to sleep on a cold, damp, mole infested lawn.

No such luck! When the Quanta Committee descended the stairs to the basement of the Bom-Banes restaurant, John Gilpin announced, with a triumphant smirk on his face, that there had been no problems at the hotel whatsoever.

The Bom-Banes restaurant has a Belgian theme and we started the Hove weekend with waffles. Would we also finish it with waffles of a different sort at the Quanta AGM?

Being a Belgian restaurant Bom-Banes was well stocked with the finest Belgian beers and we were spoilt for choice. The QL's Belgian beer specialist was not in our midst. (Where were you in our hour of need, François?) However your editor soon showed off his epicurean talents, and sat down with his carefully chosen beer. Tony Firshman immediately purloined it for a rigorous sniffing session and examined the size and shape of the glass in great detail. He then subjected the bottle and its label to a similar in depth inspection before pronouncing his satisfaction and ordering the same. Later we were joined by Per Witte.



This was a show dinner that separated the true connoisseurs from the general riff-raff. Most present preferred to drink inferior French wines and one Quanta committee member sank to even lower depths by drinking, dare I write it, Stella Artois.

Starters gave the assembled southerners a deep shock with the discovery of three savages in their midst who had ordered black pudding. Even more eyebrows were raised by the fact that two of the three were your reporter and Quanta's chairman. Was this a harbinger of a new era of peace and harmony in the QL community?

The main course had both vegetarian and non-vegetarian options and both the meat eaters and non-meat eaters praised the quality of the cooking. Dessert was a choice of waffles or ice cream.



The proprietors of Bom-Banes are accomplished musicians and provided the post-prandial entertainment. Your reporter will not even attempt to describe this in detail because it took place well past his bedtime. If he did you would accuse him of having overindulged on the Belgian beer. Suffice it to say that it had a saw used as a musical instrument; songs sang backwards; and a woman with a revolving goldfish bowl on her head. (Am I really writing this?)

The Sunday workshop followed the pattern of last year's QL shows. The attendance was low - there were 15 people at the Saturday dinner, 25 plus at the show, and 15 at the AGM. However most present enjoyed the show and there was a great deal of mixing and numerous animated conversations.

Honoured guest was Don Abbott, Quanta's new nominee webmaster, who kindly gave me a private preview of the new website. He also be-

RWAP Services NEWS!

**** We have moved ****

See our updated address details below.

We have also acquired more brand new Sinclair QL membranes and another stock of Epson Stylus Colour 850 inkjet printers, so if you need a better printer for your QL, give us a shout.

More news is always available on our website: www.rwapsoftware.co.uk

We are also looking to produce some new hard disk interfaces for the ZX Spectrum and have a few little projects on the drawing board.

Our websites:

<http://www.rwapservices.co.uk> (General site)

<http://www.rwapsoftware.co.uk> (Sinclair computer second hand and new items)

<http://www.rwapadventures.com> (Adventure Programs)

<http://www.internetbusinessangels.com> (Guidance on setting up online businesses).

New Products!

QWORD 1.0

**NOW WITH DIGITAL
SOUND ON QPC2!**

The wait is now over! Q-Word version 1 is finally available!

Platforms:

QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word £20.00
All versions with P-Word £30.00

Notes:

Q-Word DOES NOT require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.

The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2



for Windows

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Of course the full QL line is still available! (See side column)

Talent Games for Windows ea. £ 10.00
(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

Games Currently Available from www.rwapadventures.com

The Lost Kingdom of Zkul
West
The Prawn
Return to Eden

Replacement Sinclair QL Keyboard Membranes

We always have a stock of brand new Keyboard Membranes (and keyboard parts) for the original Sinclair QL, so if you have some keys which no longer work, just give us a call.

Cost is only £18.50 plus £2.75 post and packing.

Second Hand Items - Huge Range Available

We stock a wide range of books, hardware and software for the Sinclair QL, Z88 and ZX Spectrum, including disk interfaces, memory expansion and microdrive cartridges. If there is anything you need - have a look at www.rwapsoftware.co.uk (or ring us with details of your requirements).

We are always happy to help.

RWAP Services

3 Dale View Court, Fulford, Stoke-On-Trent, Staffordshire ST11 9BA TEL: (+44) 1782 398143 From the UK Dial: 01782 398143

Website: <http://www.rwapsoftware.co.uk>

Email: sales@rwapsoftware.co.uk

We Accept Payment using:



(Cheques in £ sterling made payable to R. Mellor)

Old Favourites!

Utilities

SBASIC / SuperBASIC Reference Manual on CD	£ 20.00
Sidewriter v1.08	£ 10.00
Landscape Printing (EPSON printers)	
ImageD v1.03	£ 10.00
3D object generator	
Q-Help v1.06	£ 10.00
Superbasic On-Screen help system	
Q-Index v1.05	£ 5.00
Keyword-to-topic finder	
ProForma ESC/P2 Drivers v1.04 for ProWeSs Printer Driver	£ 8.00

Applications

Flashback SE v2.03 (upgrade only)	£ 5.00
Database	
QL Cash Trader v3.7	£ 5.00
Accounting/Finance	
QL Payroll v3.5	£ 5.00
Accounting/Finance	
QL Genealogist v3.26	£ 20.00
Genealogy	
Genealogy for Windows	£ 50.00
QL Genealogist to Windows version upgrade	£ 25.00
QL Cosmos v2.04	£ 5.00
Planetarium	
Q-Route v2.00	£ 25.00
Route Finding	
Upgrade from v1.xx	£ 5.00
Britain map v1.11	£ 2.00
BIG Britain map (needs 2Mb) v2.03	£ 5.00
Various Britain Area maps (ask for details)	ea. £ 2.00
Ireland map v1.00	£ 5.00
Belgium map v1.01	£ 2.00
Catalonia map v1.03	£ 2.00
P-Word UK English Dictionary (500.000 words!)	£ 15.00
Dictionary	

Leisure

Return to Eden v3.08	£ 10.00
Adventure	
Nemesis MkII v2.03	£ 8.00
Adventure	
The Prawn v2.01	£ 8.00
Adventure	
Horrorday v3.1	£ 8.00
Adventure	
West v2.00	£ 5.00
Adventure	
The Lost Kingdom of Zkul v2.01	£ 5.00
Adventure	
All 6 games above	£ 25.00
D-Day MkII v3.04	£ 10.00
Strategy/War Simulation	
Grey Wolf v1.08	£ 8.00
Graphical Submarine Simulation	
War in the East MkII v1.24 (upgrade only)	£ 5.00
Strategy/War Simulation	
Open Golf v5.20	£ 8.00
Sports Simulation	
QuizMaster II v2.07	£ 5.00
Quiz	
Stone Raider II v2.00	£ 5.00
Arcade Game	
Hoverzone v1.2	£ 5.00
Arcade Game	
Deathstrike v1.5	£ 5.00
Arcade Game	
Flightdeck v1.0	£ 10.00
Flight Simulation	
All 6 games above (Open Golf, QuizMaster II, Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00

Notes on Software requirements

The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

came a new QL Today subscriber. I also had a chance to talk to Hugh Rooms, author of the QL Today GPS articles. He is currently developing SuperBasic programs on the use of map projection to accurately plot latitude and longitude. To help him understand and master the theory he has built some models that he had with him. Hugh hopes to write about this for QL Today and also more about the technical side of GPS.



has little experience of the QL community, because his interest in QL architecture comes via the OPD. He joined Quanta and accepted nomination to the committee to become webmaster. At the moment he is on a steep learning curve to familiarise himself with the people and traditions in the QL community.



I do not intend to describe the new website in any detail as we hope to have a full report in our news section. Suffice it to say that Don impressed most present by his willingness to listen to members' opinions and ideas. He clearly takes a flexible view of the website he is creating, and appears to be prepared to modify it in the light of experience and possible future developments.



Your reporter is a confirmed emulator user and usually shows little interest in the QL bric-a-brac available at shows. On this occasion, however, he spotted a boxed version of the Supercharge compiler complete with software, lenslock and manual. As it happens Rich Mellor was looking for just that for a client who collects lenslock programs.

Several activities were planned for the show, but as these started late there was no time for all of them. The greatest interest was in Don Abbott's demonstration of the new Quanta website. Don



The next activity was a debate on emulators versus hardware, but this did not live up to the advance publicity. This promised "Roy Wood in satin boxers" and "Tony Firshman with a bone through his nose". Your reporter was looking

forward to the photographic opportunities this would give. Not I stress of Roy Wood - QL Today is not that sort of magazine - but of Tony Firshman with a bone through his nose. It would have been a masterpiece that would have graced Dilwyn's calendars for several years to come.

The debate itself did not reach any conclusions. If anything, it showed the sheer diversity of the QL community. All too often we tend to think of our own little QL circle as being the norm, but this is far from so. Those of us who attend shows are a small minority in Quanta and we know little of what the others do and think. Quanta's 2004 survey showed that many UK QL-ers still have a basic use of the QL. An example was given of a successful businessman who still uses abacus on his QL and who has no desire to move on to a PC. None of us can say our use of the QL is superior to another man's.

Your reporter was also billed to give a presentation, but by now time had run out and Quanta's AGM was due. One member muttered to me, "How long will this take?" obviously fearing that a weekend that started with edible waffles would end with indigestible waffles of a different sort. Once the meeting started that particular member was soon one of many active participants.



The first item of any AGM is the minutes of last meeting, and here your reporter had a special interest as the incorrigible, maverick, troublemaker at last year's AGM. He had been impressed by these minutes, which had reported accurately and fully what he had said. A lesser secretary than Sarah Gilpin would have fudged or omitted some of the content. He proposed a vote of thanks for the high quality of the minutes, but strangely no other person backed this up. Why do Quanta members find it so difficult to thank their committee when they do something well?

Your reporter had another interest in the minutes. At the Manchester AGM he had raised the question of an inaccuracy in article 9.1 of the published constitution and the chairman had firmly, but incorrectly, told him he was wrong. Your reporter waited to see if the chairman would use this opportunity to apologise to both him and the members for his mistake. Of course, he did not. Why do Quanta's committee find it so difficult to say sorry when they do something wrong?

During the chairman's report there was a surprise announcement that Quanta had given a loan for the manufacture of 200 keyboard membranes. The chairman also outlined the situation regarding QMAC. Quanta can purchase the licence for about £100 but buying the source code was a bigger problem because the cost would probably be out of proportion to the use that would be made of it. There would be an initial search expenditure of about £2000. Further enquiries have yet to be made to determine if the source code is available by other means. Quanta is also unable to make any decisions about sponsoring a software author to write drivers for a possible USB device. More work needs to be done to determine its feasibility and then there would have to be a request for financial support. A further complication is that experience indicates that the programming challenge is a bigger motivator than financial rewards for our top programmers.

Figures released at the AGM gave the current size of Quanta as 169 UK members and 27 overseas members, a total of 196. This is a fall of 23 members compared with last year. However these figures should be treated with caution as the renewal forms were sent out late this year and subscriptions are still coming in. The membership secretary is shortly to write to 75 ex-members to determine their reason for leaving Quanta, although this operation has to be done carefully as many UK QL-ers are elderly. Quanta does not want to cause distress to possible widows.

No elections were necessary for Chairman, Secretary and Treasurer as they now serve a three year period of office. There were two nominations for committee posts, but the chairman reported that Roy Brereton had withdrawn his nomination because of work pressures. Don Abbott was therefore elected without a vote and welcomed as Quanta's new webmaster.

The meeting then moved on to a special resolution to amend the constitution. This increases the number of members required to petition for a special general meeting from 5% (currently about

10 members) to 20. In addition they would have to pay £300 up front. The idea is to discourage frivolous motions. Several members were unhappy about this motion, not because they disagreed with it, but because they felt that £300 was inadequate to deter a frivolous motion and that £600 or more was more realistic. (£300 not an adequate deterrent? Do Quanta members light their cigars with £20 notes.)

One member wished to propose an amendment to the motion to this effect, but was told this was not constitutionally possible and that further amendment would have to wait until next year. (This reinforces the point your reporter has been making recently that the Quanta constitution gives too much power to the executive and too little to the members. The committee can amend a member's motion, but the members cannot amend a committee motion.)

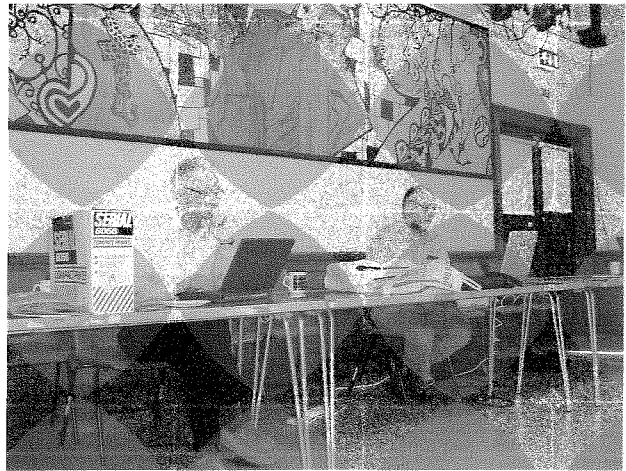
The motion was carried with just two votes against.



During the final open discussion your reporter suggested Quanta needs a news reporter. News coverage in the Quanta Magazine is very patchy, mainly because the acting editor is overworked. Also Quanta may soon wish to put some news items on the website, and Quanta also needs to improve its public relations. (For example, the chairman promised to publicise the keyboard membrane decision to both members and non-members, but did not tell us how he proposes to do the latter.)

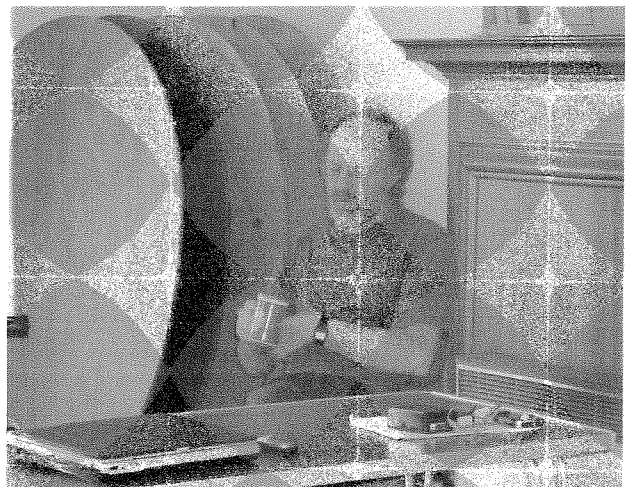
QL Today has built up a good deal of expertise on news coverage over the years, including the selection of stories, summarising mailings and sourcing illustrations. Both Quanta members and its committee are nervous of QL Today's news coverage and your editor's offer to share this expertise was greeted by cynical laughter. The offer is still there, but only for the mechanics of the job. The content is up to Quanta. QL Today welcomes healthy competition.

The Quanta weekend began with waffles, but did it end with waffles? In your reporter's opinion this was probably one of the best Quanta AGM weekends for some time. Both Don Abbott's website presentation and the AGM itself had a high level of member participation and this is something Quanta should encourage.



Quanta now has an interesting committee. Although it is still too small in numbers, it is filled with hard workers and probably for the first time in years has no dead wood. This promises well for the future, but why is "sorry" still the most difficult word to say?

A final word of warning. In the past many southerners have feared and whispered that Manchester were planning to take over Quanta. After years of patient plotting they have now achieved their aim. Three quarters of the committee are Manchurians, and their ruthlessness knows no bounds. They are even forcing the chairman to eat black pudding! Mark my words! Within a year they will make us all keep 't coal in 't bath.



Programming Assembler - Part 18a

by Norman Dunbar

In part 18 of this long running series, I showed you a small routine to read a string of ASCII characters, convert them to a floating point and then truncate that number to a long word in a register. The code in convert assumes a buffer, pointed to by A0.L, holds a string of ASCII characters without a leading QDOS string's length word. Unfortunately, most of QDOS relies on there being a length word at the start, so we really should allow for this in the convert code as well.

Well, I've been thinking (a rare thing for me – ask my wife!) and I realised that, internally, QDOSMSQ allows D7.L to be zero or the address of the first byte in memory AFTER the last character of the ASCII to be converted to a floating point value. We can use this in our favour. The conversion stops when the address in D7 is reached as QDOSMSQ loops around converting each character from the buffer.

With a slight modification to the code, we can cater for both formats of buffers – one without a leading size, and one with. The changes required are simple.

Add the following code just before the code at convert:

```
convertq  move.w    (a0)+,d7          ; Get the length word
           ext.l     d7              ; Sign extend to a long word
           add.l     a0,d7          ; D7.L correctly set, A0 also.
```

Then, remove the following line from near the start of the convert code, it's just above the call to cn_dtof:

```
moveq     #0,d7                    ; For CN_DTOF
```

So, your codefile should now look like this:

```
convertq  move.w    (a0)+,d7          ; Get the length word
           ext.l     d7              ; Sign extend to a long word
           add.l     a0,d7          ; D7.L correctly set, A0 also.

convert   movem.l   d2/d7/a1-a3,-(a7) ; Save workers
           suba.l   a6,a0            ; Relativise buffer address
           suba.l   a6,a1            ; And the maths stack
           moveq    #0,d0            ; Assume no errors
           moveq    #0,d1            ; Zero result
           move.w   cn_dtof,a2       ; Convert ASCII to an FP number
           jsr     (a2)              ; Do conversion
           tst.l   d0                ; OK ?
           bne.s   restore          ; No, bale out.
```

And that's all there is to it. You can now call the 'convert' code with A0.L pointing at a buffer of ASCII characters and no QDOS length word as long as the buffer has an 'invalid' digit at the end, a linefeed perhaps, or, you can point A0.L at a proper QDOSMSQ string's length word and call the code at 'convertq' instead.

A small test harness for the new version would be as follows:

```
test      bra.s     test2

result    ds.l      1                ; One long word for the result
           ds.b      1                ; One byte for the terminator

fp        dc.w      10                ; How long is the text ?
           dc.b      '1234567.89'    ; The fp number in Ascii plus an invalid character
           dc.l      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ; 15 Long word for a maths stack
msp       equ       *                ; This is where A1 needs to point, the STACK TOP
```



```

test2  lea    fp,a0          ; Buffer holding Ascii
        lea    msp,a1       ; Top of maths stack
        bsr.s  convertq    ; Convert from ascii to long
        lea    result,a1
        move.l d1,(a1)+    ; Save result
        move.b (a0),(a1)   ; Terminator
        rts

        in win1_source_convert_asm ; Load in the utility code

```

The above is remarkably similar to the test harness I provided in the actual article. The only difference is that the ASCII buffer at label 'fp' has been converted to a properly formatted QDOSMSQ string with a leading length word added and the 'x' has been removed from the end of the original ASCII buffer. Note the call to convertq rather than convert. Have fun.

Letter-Box

George Gwilt writes: Comments on Programming Assembler - Part 18 by Norman Dunbar (QL Today Vol11 Issue 4)

I have a feeling that Norman will be expecting me to comment on his program for converting an ASCII string to a long word in a register. So here goes.

1. On page 56 the fourth instruction of "convert" (moveq #0,d0) is not needed since the call to cn_dtof will set D0 anyway.
2. On the same page the instruction just above "restore" is:

```
adda.l    #4,a1
```

In the first place addq.l will be shorter and quicker than adda.l. In the second place the instruction is not needed anyway since A1 is reset in the very next instruction.

3. A more serious point refers not to the routine itself but to the method of checking it by using CALL. Any machine code obeyed as a result of CALL or forming part of a keyword must expect to be obeyed as part of SuperBasic. Here is what Adrian Dickens says in his QL Advanced Computer Guide. "Remember at all times that the SuperBasic program area is liable to move at any time" and "A6 and A7 are liable to change at any time". This movement can happen as a result of other programs being started up or stopped. the change in A6 and A7 can therefore occur between any two instructions in SuperBasic and so between any two instructions in a keyword or CALLED routine.

If Norman's code is to be used in an assembler executable program or is guaranteed to be used only inside a compiled program whether CALLED or as part of a keyword, then there will be no problem. Otherwise there might be.

Here is my ABC solution to this.

A. Set the ASCII string to the Basic Buffer before entry to the routine and set the relative pointer to that buffer in A0.

This can be done, assuming D2W contains the length of the ASCII buffer, as follows.

```

BV_BFBAS EQU    0
        MOVEA.L BV_BFBAS(A6),A1    ; Pointer to buffer rel to A6
        BRA.S   LOOP1
LOOP    MOVE.B  (A0)+,(A6,A1.L)
        ADDQ   #1,A1

```

```

LOOP1   DBF      D2,LOOP
        MOVEA.L  BV_BFBAS(A6),A0      ; Set A0 as needed

```

B. Use the actual SuperBasic arithmetic stack. This can be done by making sure that there is enough space on the stack and then making A1 point to it (relative to A6).

```

BV_RIP  EQU      $58
        MOVEQ    #30,D1                ; Number of bytes needed
        MOVEA.L  BV_CHRIX,A2
        JSR     (A2)
        MOVEA.L  BV_RIP(A6),A1        ; Set A1 to stack rel to A6

```

C. Inside the routine itself delete the instructions

```

SUBA.L  A6,A0

```

ZIP and UNZIP - Part 2

by Dilwyn Jones

The first part of this short series should have taught you how to use the Zip and Unzip programs on a fairly basic level.

What you may have gathered by now is that Zip and Unzip are not particularly user friendly - you need to type in cumbersome command lines and remember some obscure comand syntax. Fortunately, there is a better way.

Archivers Control Panel

Thierry Godefroy has written an extremely useful and quite simple to use front end program for Zip and other archivers. It's called Archivers Control Panel (ACP for short) and it's free. QL Today intends to include a copy on a cover disk, or you can always get the latest version from most QL PD libraries and many QL-related websites.

Archivers Control Panel is available in both French and English language versions. It requires pointer environment and a slightly modified version of the QLiberator compiler runtime extensions file, called QLIB_RUN336mod, which is supplied with the program. The modification to QLIB_RUN corrects a small problem in the original version.

Archivers Control Panel can make use of the Menu Extension from Jochen Merz, but it can also work without it. It just means the difference between having to type in filenames or selecting them from a menu.

The Archivers Control Panel package consists of just three files:

QLIB_RUN336MOD - the modified version of the QLiberator runtime extensions.

ACP_OBJ - this is the program itself.

ACP_HELP - a help file which can be viewed from

within the program, or (since it's a simple plain text file) can be loaded into most editors or printed simply by means of a copy command to send it to the printer:

```

COPY_N f1p1_ACP_HELP TO SER1
OR
COPY_N f1p1_ACP_HELP TO PAR

```

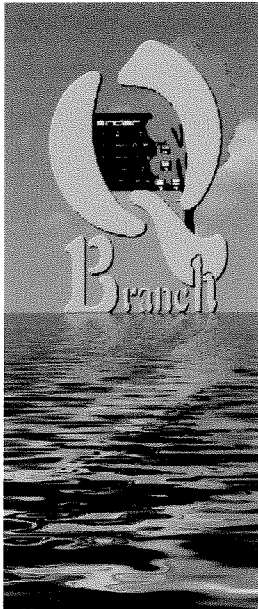
if you have a printer connected to a PAR parallel port.

Since Archivers Control Panel has a very large range of options and commands, you are strongly advised to read the instructions contained within the ACP_HELP file.

Setup

Copy the three Archivers Control Panel files into the required drive and directory on your system. This can be floppy disk such as FLP1_ or a subdirectory on a hard disk such as WIN1_ACP_. In addition to the Archivers Control Panel files, you should also copy into the same place any of the required Archivers programs. For example, if you intend to use Zip and Arc, you should copy the programs called Zip, Unzip and Arc onto the same drive or directory. If you are using the QL Today cover disk, all the required archivers are on that disk, or you can download a file containing the required archivers from Thierry Godefroy's website.

Although it is good practice to include all the archivers (zip, arc, lha, lhq, tar and zoo) just in case you decide at a later date that you need more of these programs than you originally



Feeling out on a Limb?
 Reach out for QBranch
 Suppliers of Quality QDOS/SMSQ products
 Hardware and Software

20 Locks Hill
 Portslade
 Sussex
 BN41 2LB
 UK

Tel: +44(0)1273 386030
 Fax: +44(0)1273 430501
 Mobile: +44(0)7836 745501

email : sales@qbranch.demon.co.uk
 www.qbranch.demon.co.uk

We can accept payment by
 Visa, Mastercard, Maestro and
 Switch. We also accept sterling
 cheques drawn on a UK bank.
 or payment direct to our bank:

Natwest
 Sort code 60-17-01
 Acc. 84534826
 BIC : NWBKGB 2L
 IBAN :
 GB34 NWBK 6017 0184 5348
 26



QDT
 The Ultimate QL Desktop
 v1.02
£41.50 + p&p
 £2.00 UK / 2.50 EU / £ 3.00 RoW
 Needs SMSQ/E & extended
 colours to run

EasyPTR v4
 Pts 1&2
 Major Rewrite!
£42.00
 Produce High Colour
 pointer-driven menus from
 SuperBASIC

Also on
 CD Rom

Text 87 £ 79.00
 High Colour Patch (all systems) £9.00
 ESC/P2 drivers £ 26.00 / Fountext £39.00
 Typset94 / 2488 drivers £29.00 ea.

SMSQ/E and QPC 2
 SMSQ/E for Gold Card / Atari / QXL / Q40
 Aurora High Colour Driver included with
 Gold Card Version
 Now Only **£15.00** + postage
QPC 2 v 3.33
 Now only **£ 42.00**
 Upgrade Prices:
 from v 3.xx Free
 from v 2.xx £13.90
 from v 1.xx £34.00

Delivered on CD ROM
 with Manual as Adobe
 Acrobat File

QPCPrint
 Print to any Windows Installed Printer from
 Your QL Programs !
£27.00

Programs upgraded to high colour version
 QMake / WinEd / QPAC 1 / QPAC 2 / QSUP / SuQcess / Disk Mate 5
 Send old master disk and £1.60 per disk to cover costs

Free!

Paid Upgrades	
QD 98 to QD 2003	£10.50
QSpread 2001 to QSpread 2003	£10.50

**Return Master Disk
 with order**

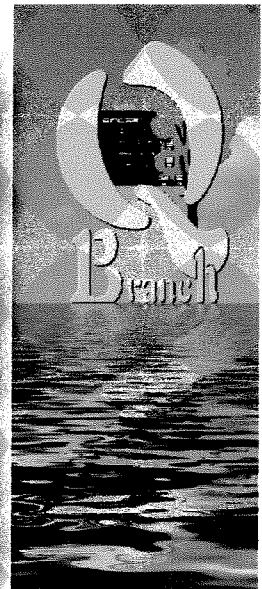
Epson EPL-6200 Printer. QL compatible Paralell Port)
with 1 new toner. 2nd user - Call for details. One only.
Big Price Drop on SMSQ/E. Any version now only

£15.00!

Coming soon. All versions of SMSQ/E on CD. We are
compiling all versions on one CD with full documentation.

Since there is extra room on the CD we also intend to
include the sources and documentation.

The new version of QPC2 is being beta tested out soon.



Utilities

Fifi2	File Finder	£21.00
QSup	Utilies	£30.00
QSpread	Spreadsheet	£51.00
Cueshell 2	File Manager	£15.00
QPAC 2	File Manager & Utilities Package	£42.00
QPAC 1	Calendar, Clock, Calculator, Sysmon	£22.00
QLoad/Qref	Fast load for QDOS	£15.00
QTYP2	Spell Checker	£31.00
QLQ	Printer Utility	£30.00
SuQcess	Database	£28.00
Q-Route	Route Finder	£25.00
Knight Safe3	Backup Program	£35.00
Fractal Collection	Fractals	£35.00
QCount	Accounting	£25.00

Programs

Programming

QD 2003	Text Editor & More	£ 49.00
QBASIC	QLiberater to QD Link	£ 15.00
QLiberator	Basic Compiler	£ 50.00
QD + QBASIC		£ 63.00
QD + QBASIC + QLiberator		£104.00
QPTR	Pointer Toolkit	£ 32.00
MasterSpy	Fast Text Editor	£ 30.00
QMake	Assembler Tools	£ 18.00
QMon/Jmon	Monitor - Upgrade only	£ 22.00
BASIC Linker	Basic Library Linker	£ 22.00
Disa 3	Dissassembler	£ 34.00
QMenu	Menu Extensions & tutorial	£ 16.00
Easyprtr v4	Toolkits & Programming Extns	£ 41.50
Easyprtr v4	Part 3 C extensions	£ 14.00

HARDWARE

We have a rotating stock of both new and second user hardware. It is best to
call or email us for details of what is available.

Recycled Items (when available)

Super Gold Card	£110.00
Gold Card	£ 45.00
Aurora	£ 65.00
QXL	£ 35.00
superHermes	£ 65.00
DiRen Keyboard	
Interface	£ 15.00
Qplane	£ 5.00

New Items

Aurora	£70.00
Aurora w/8301 & Minerva	£80.00
8mb Rom Disq	£98.00
4mb Rom Disq	£65.00
2mb RomDisq	£39.00
mPlane	£34.00
MCplate	£ 6.50
Various braQuets	£ 8.00
Gold /Super Gold Card Batteries	£10,00

We also have a collection of
standard QLs, QL Power
supplies and some QL books.

Cables for the Aurora, Qubide
and Super Gold Card ROMs
and other QL accessories are
also available from us.

Call for details

thought. For example, if downloading programs from Thierry Godefroy's website or Tony Firshman's bulletin board, you will find plenty of examples of QL programs there archived with programs other than zip - especially LHA, LHQ and ZOO.

Configuration

Archivers Control Panel has a level 1 configuration block built in, so that you can preset a whole range of options for the program. These settings are altered with the usual Config program, like most pointer driven programs.

The program can be made aware of the DATA_USE and PROG_USE settings, so it is usually good practice for these to be set to the required directory before you start Archivers Control Panel:

```
PROG_USE FLP1_
DATA_USE FLP1_
EXEC FLP1_ACP_OBJ
```

The rule of thumb is that the right hand files listing window shows the files on the DATA_USE default drive, and the archiving programs are loaded from the PROG_USE default drive.

But before you get as far as starting the program, it is a good idea to run the Config program first to configure the program prior to use.

ACP has several sections in its built in configuration block. Since there are so many items to configure, the author has split them up into groups. Once you become familiar with which section a particular item is in, you can skip whole groups to get to that one.

So execute the Config program and tell it to load the ACP_OBJ program.

At this stage it would be a good idea to read the ACP_HELP help text file for more detail on the configuration process, as it is one of the more complex parts of using Archivers Control Panel.

One of the main things you need to know is that if an at symbol ("@") is placed in front of a filename, that is used to indicate "insert the prog_use default device here. For example, if you want to load the help file ACP_HELP from the PROG_USE default drive, you'd specify its filename as @acp_help

The configuration options are as follows.

The first group of options:

- Name of the help file. This is called ACP_HELP and would normally be in the same directory as the Archiver Control Panel program itself. In common with many

configuration options, begin this entry with an @ symbol to indicate it's to be loaded from the PROG_USE default device.

- Default archive path. This is the drive where ACP looks for the zip files, zoo files and so on. Leave it empty and it will look on the DATA_USE default drive.
- Current directory. This is the drive where files are compressed from and to. Leave it empty for the program to use the DATA_USE default.
- Temporary files directory. Programs like Zip need to create some temporary files somewhere. You can either specify a specific drive such as RAM1_, or leave it empty and it'll use the DATA_USE drive.
- Sort filenames in the archive window. This will either give you a sorted list of files in the zipped archive or the files can be displayed in the order in which they already exist in the archive.
- Sort filenames in the Current Directory window. The right hand window in ACP shows a list of files on the drive from which files are compressed and to which files are extracted from an archive. Sometimes it is more convenient to have the list of files sorted than to list them in the order in which they're placed on the drive.
- Default size for archiver window. This can be BIG or SMALL.
- Keep archiver messages history. This can be YES or NO.
- Use Fileinfo II (if present) to "execute" files. This can be YES, NO or QUERY, with QUERY meaning that you are asked before it tries to use Fileinfo II to execute a data file. Files like Quill DOC files cannot normally be "executed" with an EXEC or similar command, but if you have the Fileinfo II software, it allows you to associate programs with given file types and will try to load that program then load the file into it, e.g. Fileinfo II can be taught that to "execute" DOC files, it ought to execute Quill, then drive Quill to load the file indicated.
- Save ACP config into environment variables on exit. This can be YES or NO. Environment variables will be a new subject for many readers - see the Environ_Doc file which goes with the ENV_BIN file on the disk for more details. Use of the environment variables is optional, but if you are a habitual user of environment variables, this can be a useful option.

- Default Archiver. When ACP starts, it highlights which archiving program to use unless you indicate otherwise. I normally set this to ZIP, because 99% of all archives I access are in ZIP file format. It can be any of the six programs ACP knows how to use.

The next section deals with the archiving programs themselves. You are asked to specify details such as the filename and path name for ACP to use to call that particular program. For example, if the program used to decode ARC files is called "ARC" and located in the PROG_USE directory, you may specify @ARC which will in essence result in ACP attempting to load it with a command equivalent to EXEC PROG\$&'ARC'

There's also a few questions specific to some archivers. For example, when ACP is using ZIP or UNZIP it needs to know if you want them to be dealt with by commands which are InfoZip and InfoUnzip compatible. Older QL versions of Zip and Unzip may not offer full InfoZip compatibility. You will need to specify if the TAR program is to be handled with GNU TAR compatibility (sorry, I don't know what that means!)

You are then asked for the directories containing GZIP, BZIP2 and COMPRESS. If these are on the PROG_USE default drive, just leave these entries blank.

The next section concerns advanced settings for ACP. The first set of questions asks if you want ACP to fix some of the known bugs in some archivers (these are listed in the help file). For example, LHQ does not normally handle the "*" wildcard correctly. It is worth reading this part of the help files, as knowing about these potential shortcomings will help you to resolve problems which may arise from time to time when using these programs.

Finally, this section asks you to specify any advanced options for the individual archivers. Leave these blank until you become more familiar with these programs.

When you have answered all the questions, save the reconfigured copy of ACP. Of course, it is supplied configured to be used from FLP1_ so unless you are going to use it from another device you may not need to reconfigure it at all, unless you want to set some of the more specialised features.

STARTING ACP

ACP is started with a simple EXEC command, e.g. EXEC FLP1_ACP_OBJ. This should bring up a screen as shown in figure 1.

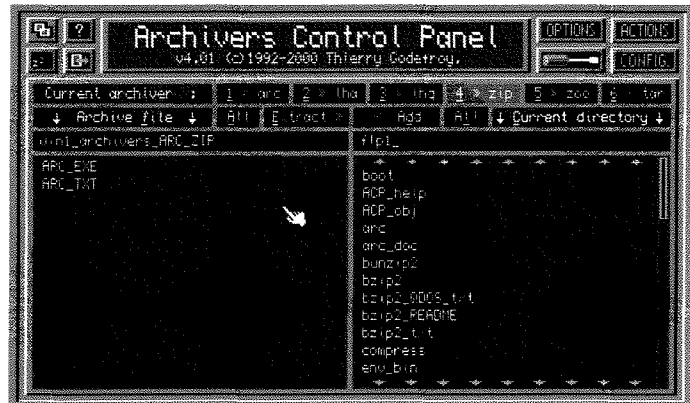


Figure 1 - ACP's main menu display

USE

You'll see at this point that it has two sets of four buttons at the top. The smaller ones at the top left of the program are (clockwise from top left): **MOVE** (the double square symbol). Moves the program display around the screen if the screen is bigger than the ACP display.

HELP (question mark symbol). Brings up the help file display, as shown in Figure 2. Here, you can press L to scroll down one line, or P to scroll down by one page. You can also click on the little icon which is a representation of a page of text with an up arrow. Press ESC to return to the main menu.



Figure 2 - The HELP screen

Zzz - this icon puts ACP to sleep in the QPAC2 button frame.

To the top right, the second group of four buttons are:

OPTIONS. This brings up the Options menu, where you can specify features of individual archivers to use. Note that this menu can vary slightly for the different archivers (the example shown is for when using Zip). It asks you to specify features such as whether the archive generated is a specifically QL-format archive, or more general for better compatibility with the same archivers on other computers. For ARC, setting this option ensures that QL file headers are stored in the archive and restored when extracted from the archive. Resetting this option allows _arc files for other operating systems to be handled. For Zip, this implies Pkzip compatibility, where the filenames stored are limited to the MS-DOS style 8.3 filenames. If handling QL files, it is best to set the QDOS Compatibility option to On. Other icons indicate if the archive created is to store the directory path names of files held in level 2 directories on a QL system. OVEWRITE allows you to specify if a file which already exists in the archive is to be replaced if you try to add a file of the same name as one which is already in the archive. A CONFIRMATIONS option allows the archiver to overwrite files automatically without asking the user, but applies only to some of the archivers. DISPLAY MESSAGES toggles whether or not the archiving program can display report messages. COMPRESS FILES lets you toggle whether files are compressed or simply stored as they are in the "real" world. For some archivers, the level of compression can be specified - it is possible to get better compression which is slower, or slightly less compression performed faster. The Encrypt Archive option is for Arc (where it controls file encryption) and Zip (where it controls password protected files - don't forget the password or key you specify here or the archive may be lost forever!)

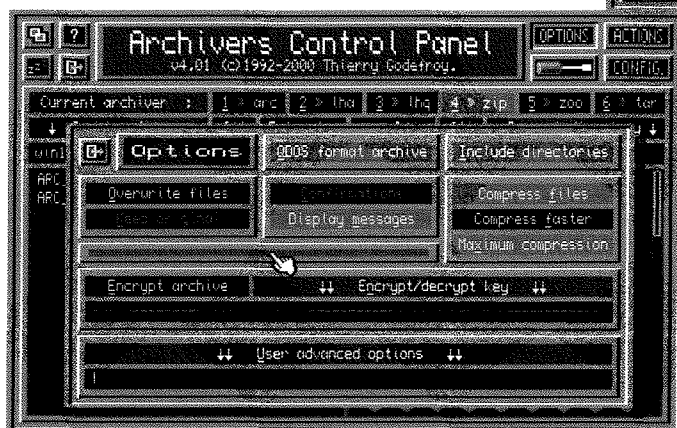


Figure 3 - The OPTIONS menu

ACTIONS. With this menu, you can perform given actions on the current archive. For example, click on the Add Files option to add the files selected in the file list window to the current archive. The Delete Files option lets you remove a selected file from the list of files in the current archive.



Figure 4 - The ACTIONS Menu

CONFIGURATION. As the name implies, this option lets you set certain options for ACP in relation to whichever of the six archivers is currently selected for use. You may prefer to leave these alone at first, until you become more familiar with the various elements of ACP and the individual archivers. Just to complicate matters, this menu varies somewhat depending on which archiver program is in use.

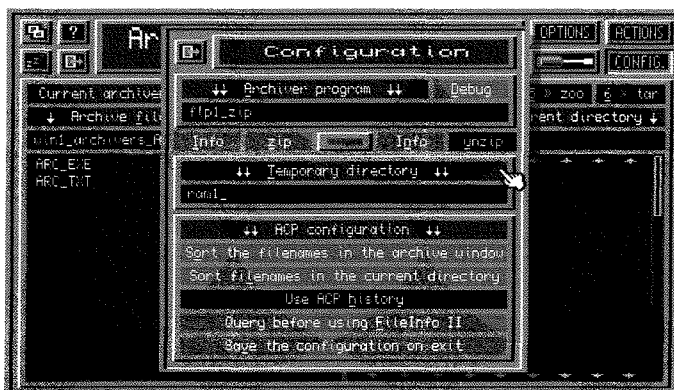


Figure 5 - The CONFIGURATION menu

TOOLS. The commands in this menu help you to use Gzip, Bzip2 and Compress on any number of individual files and to split or glue back together a single large file which is split into sections (e.g. large files too big for a single floppy disk). This is rather specialised, so I suggest you leave this alone until you have mastered basic use of ACP and then to read the instructions on using the Tools menu. Below these small buttons, there is a row of

six buttons which specify whether ACP is to use ARC, LHA, LHQ, ZIP, ZOO or TAR archivers. As most archived files on the QL scene seem to use Zip, you are probably better off learning to use ACP with Zip at first, then try out the others as your confidence grows. To change the selected archiver, just click on the name of the one you require.

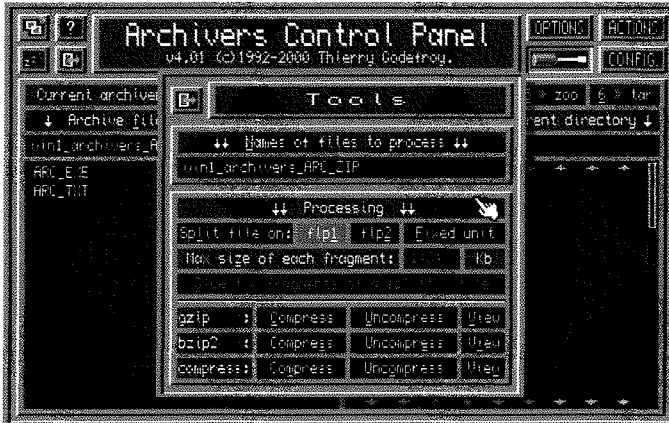


Figure 6 - The Tools menu

Below these are another set of four buttons, All, Extract, Add, All. The left hand All button selects all files in the currently selected archive, and the right hand All selects all files displayed in the

directory window. Clicking on the same icon a second time reverses the action - if all filenames were selected originally, all are deselected.

The two short and wide windows hold the name of the current archive (e.g. a _zip file) in the left hand box, while the right hand box shows the name of the current drive and directory. Clicking on one of these boxes brings up the usual file selection menu if you have the Jochen Merz Menu Extension (a file called MENU_REXT), or simply asks you to manually type in a filename in one of the boxes if not.

The two large black windows at the bottom are used for showing a list of filenames. The left hand window shows what's contained in the archive, while the right hand window shows the filenames in the currently selected drive and directory. The operation of these two windows is quite simple - files from the right hand window can be compressed into the archive file shown on the left. For example,

if you are adding files from FLP1_ into a zip file called RAM1_TEST_ZIP in RAM1_, the left hand box would show a list of files in RAM1_TEST_ZIP and the right hand box would show the list of files on FLP1_. The display is the same whether you are adding files into an archive, or extracting files out of the archive file.



JUST WORDS!

OUT OF BUSINESS!



"There's not much fun being a pirate these days. Not that there were ever that many of us in the QL community. But now trade has completely disappeared.

It's all them terrorists around. Yes, terrorising we poor pirates just trying to make a simple living. You see those people at Just Words! have stopped selling their software. Too busy editing magazines and training moles, so I hear.

Now you can download all their programs for free from the internet, or buy them for just a pound at shows. And they say their programs are now disappearing like hotcakes.

Guv, can you spare £300 for a poor pirate? I desperately need to make a frivolous motion."

Geoff Wicks, 5b Wordsworth Avenue, Sinfin, Derby, DE24 9HQ, U.K.

Tel: +44 (0)1332 - 271366

email: gwicks@beeb.net

Web: <http://members.lycos.co.uk/geoffwicks/justwords.htm>

Selection of files is easy - just click on the names of the files required. You should use the left button on a mouse, or SPACE if using the keyboard. Look at the diagram in Figure 1 - we are working with an archive called win1_archivers_ARC_ZIP in the left hand window, and the QL Today cover disk in FLP1_ shown in the right hand window. Suppose we wish to add the boot program into ARC_ZIP - we would just click on the "boot" filename in the right hand window, then click on the Add button to send a copy of "boot" into ARC_ZIP. Multiple files can be selected - try clicking on "boot", "acp_help" and "acp_obj" and send copies of all three into the archive. If you do the same thing twice, depending on configuration options which have been set up, the archiver will either overwrite the original, query whether to overwrite, or perhaps create a duplicate entry in the archive, which is not always useful of course. To extract files from an archive, just click on the names of the files to extract and click on the Extract button which will decompress the files and send them to the drive and/or directory indicated on the right hand side. Some archivers can be set to delete files as they are extracted - thankfully, Zip does not normally do this, as it is all too easy to lose files by careless mistakes when learning to use these programs!

To view a file in the right hand window, just select the names of all files to view, then click on a filename by pressing the right mouse button (or ENTER if using the keyboard). This will ask if you wish to process the files with File Info II. Reply Yes or No - if you reply with N it will try to view the file with its own built in text file viewer.

To view a file from the archive in the left hand window, just select the name and it will try to display that file in its built in text viewer. Trying to view an executable program will normally show all sorts of rubbish characters on screen, although it can be useful sometimes if you are unsure of the content of a file from its name.

And that's really all there is to using ACP at a basic level. Just remember to select the right program from the list of six, select the archive file in the left hand window, and the drive to which or from which files are to be archived from or de-archived to, select the files required then click on Extract or Add as required.

Unzip Librarian

This is a much more basic affair than ACP, but it still has its uses for those who are rather put off by the sometimes bewildering number of options and raw power of ACP.

As its name implies, it is purely intended to be used with Unzip and Unzip. It makes no use of extended options in Unzip, its purpose is simply to simplify unzipping files.

It's available in two versions, one for pointer environment users, called ULIB_OBJ, and a pointer driven version called ULIB_PTR_OBJ. Those versions have no compiler runtimes included, so versions with names ending in RTM are slightly longer, but have the compiler runtime extensions built in.

Pointer Version

See Figure 7. This has a list of drive names across the top, and a list of buttons 1 to 8 for selection of drive numbers. The <- icon lets you go back down one directory level on a system which has sub-directories. Alternatively, click on the F2 box to manually type in a drive and directory name. The large window in the centre shows a list of files in the current directory. In this list, click on the name of a zipped file and the files will be unzipped to the drive/directory shown in the F3 box at the bottom. The F4 box contains the location of the unzip program - if yours is on FLP1_ you'd change what's shown to FLP1_ otherwise Unzip Librarian would not be able to find the Unzip program to do the unzipping of files.

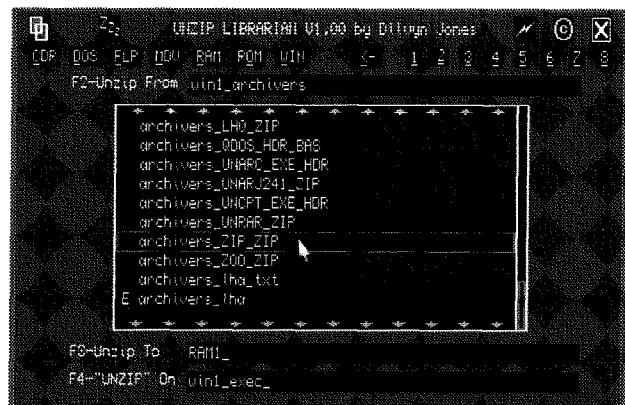


Figure 7 - Unzip Librarian, pointer driven version

Non-Pointer Version

See figure 8 for a screen dump from this version of the program. The controls are very similar - press F1 to specify where the Unzip program is stored on your system, F2 to enter the name of the drive and directory where the zipped file is stored, and F3 to enter where to unzip the stored file to. Press F5 to force a read of the selected location. <- lets you go back down one directory level. Pressing F4 redraws the display if required.

In the files window, select the zip file required and press ENTER on its filename. Sub-directories are indicated by a 'v' character before the name - press ENTER on a sub-directory name to enter that directory.

Press ENTER on the filename of the zipped file and the decompression process begins.



Figure 8 - Unzip Librarian, non-pointer version

Zip Manager

This is a program currently in development and not yet ready for release at the time of writing, but which may prove useful to users of Zip and Unzip once it is completed. See Figure 9 for a sample display.

It's laid out like my Q-Trans program, with the zipped file content shown in the left hand window and the content of the to/from drive/directory in the right hand window. There are buttons to select All or None files in that window. There's a Sort button for sorting the list of filenames. There are buttons to control encryption (password protection of zipped files), storage of directory path names and whether or not to compress files contained in sub-directories of the current directory. The commands along the bottom let

you display the catalogue details of a given entry, delete an entry from a zipped file, verify a file or view the content of a file. Adding files to a zipped file or extracting a file is accomplished simply by clicking on the appropriate arrow between the two windows.

Zip Manager has a resizable display so that you can make the file lists bigger on high resolution displays, and takes its colour scheme from the current system palette, like most of my recent programs. As a result it will only work on systems with Window Manager 2, that is, a fairly SMSQ/E or QDOS with pointer environment version 2.

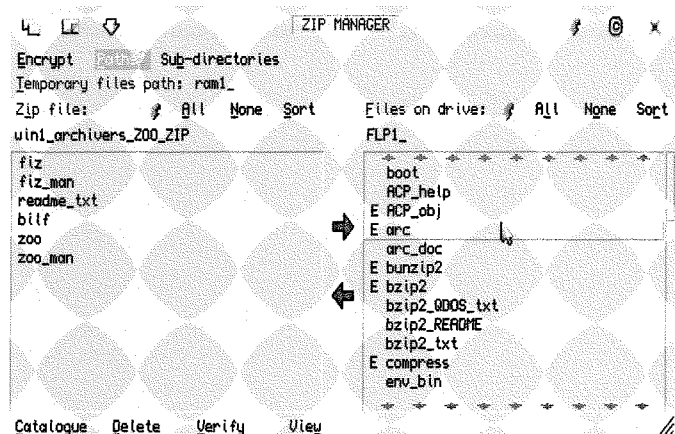


Figure 9 - Zip Manager program

Conclusions

Zip and Unzip are excellent programs, whose use can be made simpler by using a program like Archivers Control Panel.

This brings the short series to an end, I hope you've found it instructive and helpful. If there are subjects you think I haven't covered or you know of another front end program I haven't covered, please let us know and I'll try to make sure I cover it in a future article.

Essential QL Software

by David Denham

Some programs you just can't live without!

Here's a little run through of my Top 20 favourite and most used QL programs. I hope this will encourage readers to write and let us all know what their favourite programs are. I'm also using this article to drop a few hints about programs I'd like to see written for the QL.

1. Xchange

Quill and its group of programs might be over 20 years old, but they are still perfectly good programs for those like me who aren't that into Windows and Linux. They're simple to use and do most of the basics quite well. Quill can be used to write letters quickly, and to generate documentation files knowing every QL user will have a copy. Archive might be thought of as a bit

slow, but it's fine for the fairly basic databases I need to use and I can write my own little programs in the Archive programming language to make it do exactly what I want. Abacus is about as simple to use as you could want a spreadsheet to be, and although I don't use Easel much it's still a handy tool for producing the odd graph, for example.

I happily use the original QL Quill and the other three programs, but I think Xchange is so much better. The individual programs seem a little faster than the originals and less likely to crash. There's also the extra facilities glossaries, task sequencing language, the ability to control and switch between the modules and so on.

2. Pointer Environment

I wrote some time ago in this magazine that "it's not rude to point" when it comes to QLs. For years, I was happy with the original QL without pointer environment, but as more and more programs appeared which were pointer driven, and everyone was claiming pointer environment was the future, I thought I'd have a go and have never looked back.

A QL feels naked if not downright handicapped without pointer environment. Mouse control, and the ease of using CTRL C to switch between programs without having to manually redraw displays means I hate using a QL without pointer environment these days.

Since I retired, I've used my QL more than ever. I really appreciate the difference a pointered system makes!

3. Qpac 2

I've never been able to pin down exactly what it is that I like about Qpac2. It's ancient by now, yet still feels modern. There's so much about it that's good that I don't think any one facility gives it that "must have" quality, but as a package it's one of those must haves as far as I'm concerned. The Files menu is probably the single feature I use most, but other things like the button frame and all the other menus for job control and the like are so useful I wouldn't be without them.

4. Q-trans

Although I never really got on with the Launchpad, I love Q-trans from the same author. Unlike most QL file handling programs, it has two windows, one showing the content of the drive where the files are being copied from, and the other window showing the content of the desti-

nation. The whole thing seems geared up to how I like to handle files, I have yet to find anything else like it.

5. Qlay 2

When I retired and started travelling a bit more, I decided to get a laptop and emulator, since I figured that transporting the QL and monitor and disk drives and printer and everything else in my camper van would take up too much space and the equipment might get damaged. Qlay is free (actually I use the rewritten Qlay 2 by Jimmy Montesinos rather than the original version) and free software doesn't come much better than this. Its rather quirky file handling took some getting used to, in fact I'm still struggling with it a bit. But it has SuperBasic and can save to hard disk, which is great until it comes to transferring them onto floppy disk to get the files onto a QL! Still, once done that's it! One of these days I'll master the tools programs for doing this, then I'll be so very happy.

6. Qascade

This little 'start menu' is so simple to use that it's fantastic. OK, it's a bit of a drag editing the qascade_rc file which defines which programs are set up in which menu, but once that horrible job is done Qascade is so convenient to use. It just sits there in the button frame, appears on demand and all you have to do is point and click to start a program. No confusing icons, just good old English words and names!

7. Zip, Unzip and Archiver Control Panels

So much QL software is available from the internet these days that a copy of Zip and Unzip is essential. Personally I use the Jonathan Hudson versions, but there are other versions available too, although the Jonathan Hudson ones proclaim themselves to be the official versions. I also use the Archiver Control Panels program which makes it easier to use Zip and Unzip.

8. Spell Crib and QTYP

Well, we could all sometimes do with a hand to check our spelling couldn't we? QTYP is a bit old, but it still does its job very well, and Spell Crib from Geoff Wicks is so simple to use and convenient for a quick check on the occasional word. It has a degree of wild carding too - you can leave out parts of a word you are not too sure how to spell and it'll list any words in its list it thinks

matches. Just don't think you can blame any spelling mistakes in this article on Geoff Wicks though!

9. Fileinfo 2

Having used this for a while, it's hard to get used to not being able to execute a Quill doc file, a QL screen and other non-executable files. Fileinfo 2 lets you associate a program with a file type, e.g. Quill with DOC files, S-Edit with plain text files and so on. The idea is that files with a given extension are loaded into a given program which is associated with that filename suffix. It's a little bit tedious to set up, but once you get used to executing the data file from Qpac 2 files or Q-Trans rather than having to execute a program and type in the commands to load the file, you soon realise what you've been missing.

10. Discover and Textidy

Love them or hate them, PCs are a fact of life. Discover makes the job of copying files to or from a PC floppy disk so easy that it ought to be a mandatory program for everyone! Discover doesn't rearrange files for different programs, so to transfer my quill doc files to a PC I use Textidy from the same author to convert the doc to a plain text file and then Discover copies the file to a DOS format floppy disk. There's another program from Geoff Wicks (I like to drop names don't !!) called QL-2-PC which can also help with converting doc files into formats suitable for transfer to another computer, must get a copy sometime.

11. Basic Compiler

It doesn't matter if you prefer Turbo or Q-Liberator, both are great for compiling your basic programs, making them into fast, self contained multitasking programs.

12. Toolkit 2

Everyone should have it. In fact, now it's available free, everyone can. It's built into disk interfaces anyway, and you can get a plug in ROM or an image for use in emulators.

13. Screen Snatcher

On a PC it's possible to press one key and a copy of the screen picture is placed on the clipboard. There's no direct equivalent on a QL, although Screen Snatcher can save a copy of the display to a file. I write a lot of articles for QL Today as you may have noticed, so a program

like Screen Snatcher is very useful for illustrating an article.

14. Snake

I'm not much of a game player (too old, that's my excuse and I'm sticking to it) which is why I hate Dilwyn Jones. I know that once I've started playing this damn game (which involves nothing more daring than guiding an ever growing and very hungry snake around the screen avoiding its own body parts) I'm not going to get much else done.

15. SPL

Not so much a program as a command which starts a multitasking job going to spool a file to the printer. Printing long files no longer stops you using the QL until printing finishes, SPL just prints in the background letting you get on with something else while that big long file prints.

16. Photon

Dave Westbury's JPEG graphics file viewer is so easy to use for viewing JPEG files, which are pretty well everywhere these days. Unfortunately, it doesn't save the converted pictures (it's only a viewer) but a few basic commands can work around that by executing Photon to view a specified file, issuing a PAUSE command to give Photon a chance to display the picture, then saving the screen with an SBYTES command. On a QL with a standard QL display, something like:
`EXEC FLP1_PHOTON; 'f1p1_sample.jpg':PAUSE
500: SBYTES f1p1_sample_scr,131072,32768`

Which should make Photon wait until you press a key to make it go away. Of course you will have to adapt the SBYTES command for SMSQ/E systems to make sure it saves the screen in the correct way, using something like:

```
SBYTES ram1_sample_scr,SCR_BASE,SCR_LLEN  
* SCR_YLIM
```

Now all we need is an up to date graphics editing program!

17. Sdump

A simple to use screen dump command built into every Miracle Systems disk interface. Makes it easy to write screen dumps into your own programs by using an SDUMP command, as long as your printer is one of the ones supported. To the best of my knowledge, sdump only works with mode 4 and mode 8 pictures.

18. S-Edit

A really useful and simple to use text editor from Germany. Not pointer driven but don't hold that against it. It has word wrap, block handling and so on and can handle pretty large text files.

19. Euro Currency Converter

Neat, simple little program which takes the brain strain out of converting between British and European monies on my travels! The work of one Andrea Carpi in Italy, I gather.

20. Line Design

Number 20 in the list, but much more useful than the position implies. So easy to make and print posters and text and graphics you can resize without losing too much detail.

And as my top 20 includes pointer environment, which isn't really a program in the usual sense, I'll add a route finder program to the list even though I don't actually own one. Some time ago I got to use Rich Mellor's QRoutes program on someone else's machine and liked it very much. I keep asking Santa for a copy as a Christmas present, without success so far.

That's my top 20 QL software, now how about letting us know about your favourite QL programs?

To close, here's my wish list of programs I'd like to see written for the QL:

1. An email program.
2. A web browser. Lynx is all well and good, but it is text only and doesn't handle secure websites. Even if the browser could only handle offline pages, that would be useful for viewing HTML stuff.
3. A program which can display and print mixed text and graphics. Doesn't have to be a full word processor like on other computers, but it is high

on my wish list. A program which lets you define text boxes and columns, and lets you position some QL pic files on the page would do just fine.

4. Screen dumps for high colour screens. I don't use them (yet!) but when I do it'd be nice to be able to print colour pictures when I invest in a suitable "QL".

5. A flight simulator. One of the things I regret never having done in my life is to pilot a plane. I guess this will be as close as I get to it!

6. Translation software. I've used a translation service on Google. You paste in text in one language and it gives you a machine translated version in another language. Hardly perfect, converting from German to English gives weird results, but at least it's sufficiently readable for understanding the gist of what was written. Very useful for converting the instructions for freeware French and German QL programs!

7. A decent file finder program. Some of these do exist, but none of them seems to do exactly what I need, which is to search based on part of a filename (possibly with wildcards), a few words contained within the file, and to create a list of files found which I can then view or load to check the results.

8. Decent Space Invaders and Pacman games for the QL. There are a few, but they either don't work on a modern system, or run way, way too fast on a Gold Card. These are the only arcade games I ever really got to like.

9. A PDF file reader for the QL. PDF files seem to be so common on the internet that a file viewer would be very useful.

10. My final wish is more for a hardware device than software, but I'll include it anyway. That's for a sound card which would let me play music and sound effects and which was easily programmed. No doubt, someone will now tell me that some of these already exist. Even better, someone will conveniently write a program just because I asked for one!

Lightning Strikes Again

by Simon N Goodwin

Norman Dunbar's long running 68K assembler programming series in this magazine, and the feedback from GWASS author and Turbo maintainer George Gwilt, often remind me of the elegance of QL machine language and the techniques and tricks that go with it.

The part of the series in Volume 11 issue 4 brought to mind the powerful 32 bit capabilities of every Qdos system, right back to the original

68008-based QL, and led me to revisit some fast long integer conversion routines, to and from ASCII string format, that complement the vectored utilities in Qdos which Norman's article exercised.

His Convert routine demonstrates two useful ROM functions - the vector to convert an ASCII numeric string into a six-byte floating point

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets. Can be used with lights/hifi/monitors—ie a QL monitor can be used as a switch control.

Cost **£24**

superHermes

A major hardware upgrade for the QL. All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/F // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software) **£90** (£92/£93)
 IBM AT UK layout Keyboard..... **£11** (£13/£15)
 Serial mouse..... **£8** (£8.50/£9)
 Capslock/scrolllock LED **£1** (£1.50/£1.50)
 Keyboard or mouse lead **£3** (£3.50/£3.50)
 High speed serial (ser3) lead..... **£4** (£4.50/£4.50)

Hermes available for **£25** (£26/£27) Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead) **£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if reqd). Send disk plus SAE or two IRCs

MK1..£40 (£41/£43) MK1L..£65 (£66/£67)

MINERVA RTC (MK1) + battery for 256 bytes ram.
 CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL. A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second. Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£40/£41)
 4mbytes RomDisq..... **£65** (£66/£67)
 8 mbytes RomDisq..... **£98** (£99/£100)
 Aurora adaptor..... **£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost **£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors)..... **£40** (£43/£44)
 4 amp total (for motors etc) **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..... **£25** (£27/£28)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting..... **£30** (£31/£32)

Temp probe (-40°C to +125°C)..... **£10** (£10.50/£11)

Connector for four temp probes..... **£10** (£10.50/£11)

Data sheets **£2** (£2.50/£3)

Control software & manual (for all I/F)..... **£2** (£2.50/£3)

QL SPARES

Keyboard membrane no longer on sale

1377 PAL **£3** (£3.50/£4)

Circuit diagrams **£3** (£3.50/£4)

68008 cpu or 8049 IPC..... **£8** (£8.50/£9)

8301/8302 or JM ROM or serial lead..... **£10** (£10.50/£11)

Power supply (sea mail overseas)..... **£12** (£19/£23)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable) Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address/postal order or CASH I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details 22 Feb 03

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: +44 (0) 1442-828254

Fax/BBS: +44 (0) 1442-828255

tony@firshman.co.uk

http://www.firshman.co.uk

number, and the one to round a decimal to a 32 bit integer.

But if you want to go directly between strings and long integers there's no suitable routine in Qdos, Minerva or SMS. That's a pity as such conversions are potentially useful and amenable to relatively speedy implementations using the native capabilities of 68K chips.

In the year between the release of Supercharge, the first SuperBASIC compiler, in 1985 and Turbo in 1987 I spent several months working on a compiler that would have been faster than either, albeit more restrictive in the dialect of SuperBASIC it compiled. This compiler was code-named LIGHTNING - a name re-cycled by Digital Precision later, after I stopped developing compilers.

Lightning was a true 32 bit compiler, with unlimited program size (breaking through the 64K limit of Supercharge) and support for 32 bit integers, rather than the 16 bit ones in SuperBASIC and all its other compilers, exploiting the 32 bit data paths common to all 68K-series processors to work with nine and ten digit values almost as fast as 16 bit code can handle those in the range from -32678 to +32767.

The aim was to produce a low-priced, very fast native code compiler (with no threaded code) for programs that could still be prototyped in SuperBASIC - as long as floating point numbers were not relied upon - but would work a lot faster than any based on software floating point, and the long library routines needed to process decimal values one bit at a time, in software.

But when Sinclair sold out to Amstrad my publisher Freddy Vachha feared that the QL market would soon evaporate, rushed out the release of Turbo Toolkit and ordered me to stop work on Lightning and concentrate on Turbo, which had been intended to be the follow-up third compiler in the series.

So Lightning was never finished, although Gerry Jackson and I recycled some of the code, including the fast memory handler and 32 bit code-generator, in Turbo (which eventually came out

roughly when it was planned to be released in any case, even though Lightning was cancelled). Some related runtime code ended up in Digital C - a 32 bit 68K native code compiler aimed at a similar market to that intended for Lightning, but without the conveniences of interpretative development and source-level debugging.

There were plans to add 32 bit integer support to Turbo, and Gerry did some work on that in the following couple of years, but work on the 32 bit integer support library stopped in 1991, as compiler royalty payments had ceased.

15 years on, part 18 of Norman's Assembler Programming column in this magazine prompted me to dig out some of the unpublished 32 bit support code. Norman's program converted from string to 32 bit integer format using two ROM routines, creating a temporary value in six byte floating point format on the maths stack in the process. While this makes good use of existing code it's a roundabout way to perform the conversion, as anyone who has studied the ROM coercion routines will realise. If you haven't but now wish you had, the JS ROM source for such conversions appears towards the end of the freely-available QDOS4.ASM disassembly, supplied with Amiga Qdos emulator.

Turbo Toolkit has been updated to include LONGINTEGER and LONGINTEGER\$ functions, but these convert between floating-point and 32 bit long integer format, rather than strings and 32 bit integers. The two routines presented here are complementary to those, as they convert between string and 32 bit integer format.

So, more than two decades after they were intended to be released, here are the Lightning library routines for conversion between 32 bit integers and ASCII strings.

The first derives a long integer value from a string. It uses no ROM calls but is exactly the same length as Norman's code (54 bytes). It converts a string of D7 bytes at the address in A0 (so the D7 parameter is D7-A0, in the format used by Norman's code when the end of the string is known) and it leaves the result in D3:

```
* Lightning codegen library, str_to_long subroutine
* Enter with d7 holding the length of the string
*           a0 pointing to the string characters
* Return with d3 holding the long value
*
* Returns 0 if input is non-numeric; result is undefined
* for input strings outside the signed 32 bit range.
```

```
str_to_long  moveq    #0,d3      d3 will hold the long integer
              moveq    #0,d0      Need ms 3 bytes of d0 to be 0
```

```

        subq.w #1,d7      For DBRA below
        move.b (a0)+,d0  First character
*
        moveq #0,d1      Sign indicator
        cmpi.b #'-',d0
        bne.s stol_not_neg
        moveq #1,d1
        subq.w #1,d7
*
stol_loop  move.b (a0)+,d0  Next digit
stol_not_neg subi.b #'0',d0
           cmpi.b #9,d0
           bhi.s stol_do_sign
           move.l d3,d2
           lsl.l #3,d3
           add.l d2,d3
           add.l d2,d3      d3 := d3 * 10
           add.l d0,d3      + next digit
           dbra d7,stol_loop
*
stol_do_sign tst.b d1
            beq.s stol_exit
            neg.l d3
stol_exit  rts

```

Comments on str_to_long, Norman's code and George's comments

When I read Norman's article in QL Today I contacted him to get the source and he forwarded the riposte from George Gwilt (printed earlier this issue). I agree with both of George's first two comments, and was planning to mention the former myself, and also that the fifth instruction (moveq #0,d1) seems similarly redundant as D1 is clobbered by CN.DTOF, and then by the actual result if the conversion succeeds and documented as 'undefined' if it fails (D0 \neq 0). So that instruction too can be removed.

As far as I recall, the requirement to clear D7 before calling RI.EXEC was only needed for the first Qdos ROMs (up to "JM") and then only if the function was RI.EXP or that was in the RI.EXEC or RI.EXECB sequence or called within the operation (as RI.EXP is typically used inside RI.POWFP, for example). So I believe it's never needed in code like Norman's, but probably safest to leave it in just in case...

I mentioned this to George, who replied: "The requirement for a zero D7 is given in the QL Technical Guide by David Karlin and Tony Tebby." But many ROM maths operations do not make any use of D7, e.g. RI.NEG, RI.DUP and RI.SWAP (just a few I was able to find source for quickly), RI.ABS and I believe RI.ADD (without checking the source in the "AH" ROM again, though I did decades ago) which is the only non-trivial routine called by RI.INT and RI.NLINT (and

still fairly simple). However I don't have all possible ROMs to hand, to test this. George checked "JS" and noted "The JS ROM certainly does not need a zero D7. It seems that all the operations start by saving D4-7 and A4."

The biggest difference between the ROM approach and the Lightning code is the speed of conversion. str_to_long is almost 16 times faster than the ROM at converting the string "123456789" into a long integer, on my Gold Card, and 21.5 times faster on the simpler - but commonplace - value "0". The signed value "-9999" is converted 11.7 times faster by the Lightning code. Speed is not everything and there are other differences between the routines; the ROM allows exponential notation, which Lightning did not; in the most extreme cases, this narrows the performance gap.

In one such case the ROM approach can parse the value "-1E9" only 6.5 times more slowly than str_to_long can convert the equivalent "-1000000000", which the ROM digests less than half as quickly as the same value in exponential format. str_to_long converts the 11-character representation to binary 13.7 times faster than the ROM approach, on my QL Gold Card QL with Minerva 1.97.

Numeric string to binary conversions are often limited by the speed of other devices - such as file access - but may take a significant proportion of the runtime in compilers and assemblers, especially when those run to and from RAM, as the fastest invariably do.

Another difference is that the ROM also rounds decimals, as the QL Today article showed in passing, but I find it rather perverse at the limits - at least on Minerva - I suspect the rounding up of 112.49999999 to 113 (but not 112.4999999, which yields 112) is because the rounding is done after conversion from string to floating-point rather than in CNDTOF, so it suffers from the inability of binary floating point to represent non-binary fractions exactly - but that's another can of worms I've written about before, several times.

My version stops at any non-numeric character and returns 0, rather than an ERRNO, if that's the first; it does not trap overflow, because in general Lightning did not; The naive way to do this would be to test the overflow flag after each long word maths operation as digits are accumulated, but this is slow and complicated by the signed result (e.g. the minimum valid input, -2147483648 has the same representation as 2147483648, which is one greater than the maximum allowed; integers may usefully be considered signed or unsigned, e.g. to simplify PEEKs and POKEs to the top half of the memory map. Indeed my Z80 compiler ZIP deliberately exploited this. So if I was going to write a version capable of trapping overflow I'd probably start by checking the number of characters, in D7, and use the existing code if this was less than 10 and a more complicated version only if the number was potentially long enough to overflow - a rare case, in general, I suspect.

The final value of D7W is \$FFFF only if all the supplied characters were consumed. This is another difference which could be good or bad depending upon the context in which the conversion is called.

Optimisations

George did suggest a faster way to implement the long multiplication by ten in stol_loop. Instead of:

```

move.l d3,d2
ls1.l #3,d3
add.l d2,d3
add.l d2,d3
add.l d0,d3

```

Which copies the running total, shifts it left three places to multiply by eight, then adds the old running total from D2 twice to scale it up by a factor of ten, George proposed:

```

add.l d3,d3
add.l d3,d0
as1.l #2,d3
add.l d0,d3

```

D0 holds the new digit, 0..9, and D3 is the running total from digits already scanned, so after the second line above D0 contains $D3 * 2 + D0$; the shift left multiplies the doubled running total by a further four (so D3 becomes the initial D3 times eight) and the final ADD adds the initial $D3 * 2 + D0$ to the initial $D3 * 8$ to get $D3 * 10 + D0$. Elegant!

This has a similar effect but in one fewer line and does not use D2. It's two bytes shorter and 10 cycles faster on a 68008 (6 faster on 68000 or 68010, probably 2 on 68020 (sped up by the instruction cache and barrel shifter) compared with the Lightning code.

This variant works fine for values up to about 10,000, but requires another change to cope correctly with larger values, wiping out almost all the advantage. It can leave the high order bytes of D0 non-zero when we return to STOL_LOOP, if the input has more than four digits.

For instance if the string is "99999", D0 and D3 will both be 9 at the end of the first pass, 27 and 72 as the second digit is added (giving 99, correctly) at the end of the second pass, then 207 ($99*2 + 9$) and 792 ($99*8$) giving the expected 999 in D3 after processing the third digit; but the fourth pass adds $999 * 2$ to the fourth digit 9 before the ASL, so the D0 and D3 values added are 2007 ($999 * 2 + 9$) and 7992, giving 9999 (hurrah) but leaving \$7D7 in D0.

So after picking up the fifth digit D0 contains \$709, rather than just 9. To this we add $9999 * 2$ (the accumulated total from D3, after doubling) getting 21799; D3 goes on to 79992 ($8*9999$) but when D0 is added the additional 21799 yields 101791 - whoops!

One solution would be to clear the high bytes of D0 every time, with `moveq #0,d0` at the top of the loop - but this means all the speed-up, other than a couple of cycles for the shorter shift on early 68K processors - is lost. But this experiment did yield a slight speed up for the Lightning code, saving a couple of cycles on 68000 and 68008 processors by reducing the number of shifts; those chips shift one bits at a time, whereas the 68020 introduced a 'barrel shifter' that can shift any number of places in constant time:

```

move.l d3,d2
ls1.l #2,d3
add.l d2,d3
add.l d3,d3
add.l d0,d3

```

Changing the penultimate instruction to add D3 to itself, instead of D2 to D3, means the logical shift only needs to move two bits left, rather than

three. Eagle-eyed readers may have noticed that George used ASL rather than LSL, arithmetic rather than logical shift; in this case there's no difference as we're shifting an unsigned value to the left, but arithmetic shifts of SIGNED values to the RIGHT preserve the sign, whereas a logical shift would be needed for a big unsigned value. You might wonder why we don't just multiply by ten, using the instruction apparently made for the purpose! It's almost always faster to multiply by ten with adds and shifts, on any 68K chip other than a 68060, because the 32 bit MUL instruction is microcoded and loops internally, taking several times longer than three or four simpler instructions even though it uses less code.

Chips before the 68020 are limited to 16 bit inputs for multiplication, so the shift-and-add approach is shorter as well as faster for them, unless you only need to convert numbers up to five or six digits long.

If you're really pushed for speed on a 68008 or 68000 processor and can spare a couple of registers, it's worth preloading them with the constants #0' and #9, using moveq, to save those values occupying 16 bits of code (and the associated data fetches) each time round the loop. This won't help on later processors because they load four-byte instructions from their instruction caches as quickly as two byte ones.

Reversing the process

Lightning also included a routine to go the opposite way - from a 32 bit integer to a numeric string. You could again do this using ROM floating-point routines but they're horrendously slow and complicated, using a chain of floating point multiplication or division operations to normalise the input and a pair of DIVU instructions to crank out each digit - those are about the slowest in the 68K repertoire, taking around 150 cycles each on a 68000 and 20-odd on a 68040 or '060.

The following routine expects the register assigned the symbolic name 'data' (e.g. A4) to point where the ASCII string is to be stored. Up to 11 bytes may be stored, from that address onwards.

As long as that space is available, no error is possible, so the number of bytes written is returned in D0.

It starts by checking for a negative value and emitting the minus sign and negating the input if it finds one. You can use it to write unsigned values to to 2³²-1 (4 billion or so) if you enter the code at ltos_is_pos with the address of the start of the output buffer in a2 as well as register 'data'.

After that the code is almost naively simple, but still quite fast. It attempts to subtract progressively decreasing powers of ten from the value in D3, counting the number of times it succeeds, until the subtraction fails. If the resultant count is non-zero or a digit has already been output (so we've passed any leading zeros, which we suppress so that "123" is not displayed as "0000000123") it converts that count into an ASCII digit and keeps going till only one digit remains - the least significant, which is always output.

This code could be faster, though more complicated, if it tested the magnitude of the initial value before starting ltos_loop - skipping the first four powers of ten in the table, say, if D3 is less than a million at the start. The snag of this sort of optimisation is that it is domain specific - it helps with some values but makes things slower if mainly big numbers need to be printed.

You could reduce the average number of subtractions to find each digit substantially by experimentally subtracting a multiple of each power of ten - say five times, the current table values - in the hope of replacing five subtractions with one when the digit was between five and nine. Unfortunately this makes the loop slower when the digit happens to be in the range zero to four. You might find best results by trial subtractions of three, three, two and one times the current power of ten, in various combinations, all aiming to detect ten cases with four or fewer test subtractions. But again the benefit depends upon the input and your willingness to trade extra code and data space for speed. Even the simple routine listed below is a lot faster than the floating-point ROM approach!

```
* Long Integer to String conversion, from QL Lightning
*
* In: D3 = value to convert, data -> buffer for result
* Returns string at data, length in D0 - no error possible
* Clobbers A0, A2, D0, D1, D2, D3
```

```
data          equ      a4          Points to output buffer

long_to_string  move.l   data,a2     Remember start of string
                tst.l    d3
                bge.s   ltos_is_pos
                move.b  #'-',(a2)+ Save negative sign
```

```

ltos_is_pos    neg.l    d3
               beq.s    ltos_exit

               moveq    #0,d2    Leading 0 indicator
               lea.l    ltos_divisors,a0

ltos_loop     moveq    #-1,d1    For digit
               move.l    (a0)+,d0  Power of 10
               beq.s    ltos_exit  Ran out of divisors

ltos_still_pos addq.b    #1,d1
               sub.l    d0,d3    Continue until negative
               bcc.s    ltos_still_pos

               add.l    d0,d3    Too far, make positive
               tst.b    d1        Is digit non-zero?
               bne.s    ltos_save_char

               tst.b    d2        Leading 0 ?
               bge.s    ltos_loop  If so, ignore it

*
ltos_save_char moveq    #'0',d2    More zeros will not be leading
               add.b    d2,d1
               move.b    d1,(a2)+
               bra.s    ltos_loop

*
ltos_divisors dc.l    1000000000  Table of powers of 10
               dc.l    100000000
               dc.l    10000000
               dc.l    1000000
               dc.l    100000
               dc.l    10000
               dc.l    1000
               dc.l    100
               dc.l    10
               dc.l    0

*
ltos_exit     addi.b    #'0',d3
               move.b    d3,(a2)+
               move.l    a2,d0
               sub.l    data,d0  Calculate string length
               rts

```

Light Relief with Strings

by George Gwilt

Some odd things can happen with strings in S*BASIC. All strings are made up of a word of two bytes giving the length, l, of the string. This is followed by l bytes. Thus, inside the computer, "Hello" appears as the following 7 bytes:

0,5,72,101,108,108,111

The reason odd things can happen with strings is partly due to the two ways in which strings can be accessed by the language. Strings can either be undimensioned or dimensioned. If you type

```
DIM a$(100):a$="Hello"
```

the string a\$ is dimensioned. When this happens it becomes an array and a "descriptor" is set up giving details of this array. This will include the number and size of dimensions. In the case of our "Hello" there is only one dimension and its size is 100. The descriptor also contains a

pointer to where the array is stored.

If a string is not dimensioned there is no descriptor.

How does S*BASIC access these strings? The variable a\$ will appear in the Name List and there will be a corresponding entry in the Name Table. This latter entry contains a pointer which is to the string itself if it is undimensioned or to the descriptor if it is dimensioned.

Knowledge of the details of the Name List and Name Table is not essential to an understanding of the quirks I am about to reveal. Nevertheless having mentioned these two entities I shall give a quick superficial explanation of them. The Name List is just what it says, a list of names. Every time you define a variable in S*BASIC the name of that variable is given a place in the Name List. At

the same time a further entry is made in the Name Table. This means that the name appearing at, say, the 201st entry in the Name List has a corresponding entry at the 201th element of the Name Table and so on. Each element in the Name Table is 8 bytes long, the second four bytes of which is a pointer to the value of the variable.

The two types of string have different effects on the speed of access to the strings. For undimensioned strings S*BASIC grabs space from the heap for storage. When a string variable is given a value of a different length, the old storage area is thrown away and a new one grabbed. Compare that with the dimensioned string where the space for the string, however many times it is redefined, is set once and for all, or until the string is redimensioned of course. It would seem that dealing with dimensioned strings is faster than with undimensioned ones. That's why Turbo requires all strings to be dimensioned. Speed!

There is another aspect. If you try to set a string larger than the dimensioned length in a dimensioned string variable the extra characters are lopped off the end without so much as a "by your leave". Thus:

```
DIM a$(4):a$="Hello":PRINT a$
will give you "Hell".
```

Pieces of String

S*BASIC allows you to extract and insert pieces of a string. In the examples above:

```
PRINT a$(2 to 3)
```

will give "el", the second and third characters of the string. You can also insert something into a string, for example:

```
a$(2 to 4)="Hmm":PRINT a$
will give "HHmmo".
```

This extraction and insertion works for both undimensioned and dimensioned strings.

Now comes something stranger. If you extract character 0 you will find that for an undimensioned string you get an "out of range" error, but if the string is dimensioned you will see the length of the string! Thus

```
PRINT a$(0)
```

gives 5 for the first example of "Hello". Also

```
a$(0)=3:PRINT a$
```

gives "Hel" because the string length has been changed.

Lets go for something odder.

```
PRINT a$(1 to 5)
```

now gives "Hello" even though we have altered the string's length to 3. We can find something about the contents of the area set aside for the maximum string as follows:

```
b$=a$(1 to 10):Print b$;LEN(b$)
```

This should print the entire contents of the 10 byte area followed by "10", since b\$ should be that long.

In fact we get "Hello 10". There are six spaces between the "o" of Hello and the length "10". This shows that the area for the maximum string is padded out with spaces.

Let's see what happens if we define a\$ as a different string. What happens if we type:

```
a$="New":print a$(1 to 10);a$(0)
```

Well, we get "New 3"

That is, we get New followed by seven spaces and then the new length, 3. This shows that for a new value for the variable a\$, S*BASIC sets all the remaining characters to spaces after the new string is entered. Thus the "lo" at the end of the previous "Hello" has been covered up.

If we redimension a\$ the contents of the string become zero bytes, and the string length is 0. As soon as a\$ is defined as some string or other, even a null string, the area is padded out with spaces.

You should now see that it is possible to use the end of the space set apart for a dimensioned string after the string itself without disturbing the string. The characters after the string itself up to the end of the area are called "unset". Jan Jones in her Definitive Handbook on QL Super BASIC says:

"If you modify an unset character, the element will be changed but the actual length of the string will not. SuperBASIC assumes that you know what you are doing."

What an opportunity this is for doing things in an unconventional way. You can reserve space up to 32766 bytes by dimensioning a string. By setting the string to a null string you have 32766 spaces. You can put any set of characters into this space and retrieve them. What if you forget where the characters are? Never mind just set the length and use INSTR. Here's an example:

```
DIM x$(32766):x$="":x$(1000 to
1004)="Where"
```

to find where "Where" is we do;

```
x$(0)=32766:PRINT "Where" INSTR x$
```

We see 1000 printed so we have found the word we stored.

This is all very amusing no doubt, but there is a snag. If you want to compile a program containing these tricks and want to use Turbo you may find the compiled program giving odd results.

Curiously enough Turbo does allow you to insert characters anywhere in the area of a dimensioned string irrespective of the current length. But trying to alter the stored length by using a\$(0) will not work. Turbo takes a\$(0) to be the same as a\$(1 TO), ie the whole string. Thus:

```
DIM b$(7):b$="Gave1":b$(0)=4:PRINT b$
```

will give "Gave" in S*BASIC but "4" if compiled by Turbo.

Also, although S*BASIC sets the "unset" characters to spaces whenever the string is redefined, Turbo leaves the unset characters at their previous values, which will be CHR\$(0) if they have never been set.

Thus;

```
DIM b$(7):b$="Gave1":PRINT b$(1 to 5): b$="Ho":PRINT b$(1 to 5)
```

gives

```
Gave1  
Ho
```

in S*BASIC but compiled by Turbo it gives

```
Gave1  
Hove1
```

Conclusion

There are two types of string, dimensioned and undimensioned. You are, surprisingly, allowed access to the whole area earmarked for dimensioned strings. This has amusing consequences, but Turbo users beware.

A Postscript

I have mentioned several differences in strings which depend on whether or not they are dimensioned. I should perhaps mention another case. I expect that it will not be generally known. In QPTR there are several keywords added to allow the setting up of a PE window working definition. One of these keywords is the function MK_LIL which produces a loose item list and returns its address. One of the parameters to be supplied is "sk\$". This is a string giving the select keys (each a character) for the loose items.

Although the manual does not specify it, this parameter, sk\$, must be passed as an undimensioned string. Otherwise an error is signalled.

Electro-Magnetic-Pulses

by Stephen Poole

You are on that beautiful blue planet Earth, up here in space. All is well until a hyper-dense red neutron-star drifts into view. As any electronics engineer will know, neutron stars emit powerful electro-magnetic interference, that will destroy all QL activity on earth. To save your Computing Community, you must get Ian Pizer and the CERN particle-accelerator to lob black holes at the dwarf red neutron star, and hit it before its pulses reach you! But the GPS system is already down, so the best we can do is to estimate the pulsar's distance! Its up to you to input its estimated coordinates and see where the black-hole evaporates... Then try again. As you try, the pulse gets nearer to you. (Artistic Licence allows me to portray black-holes in shooting-star green. Black holes are not strictly black: They give off enormous quantities of radiation as they tear matter apart!). According to the difficulty level you have chosen, you will be allowed a limited number of tries, with level one being the easiest. As always with the coordinate-system, x or y may be posi-

tive or negative, but to make things a bit easier for you, your last estimates remain on the screen until you fire again. If you guess x or y correctly, you are only cued to INPUT the remaining coordinate.

Although there are approximately sixty lines of code to type in, this is necessary because such games need a high level of input error-trapping, to avoid the frustration of the game halting on a bug. The ZX80 program that originally inspired me, had but 25 code-lines. But that game was about terrorists leaving radioactive parcels in busy public places, so I rewrote an entirely new game from scratch, a game which is much more to my tastes. This game will develop your sense of precise estimation of distances at varying scales. But it will also keep you reminded that it is the environment which poses the biggest threat to mankind. Dealing with people is an altogether more trivial task! Or so it should be... Happy black-hole generating!


```

110 REMark EMP_bas, by S.Poole. v21sept2006 (for QL Today)
120 CLEAR: OPEN#1,con_16: WINDOW 512,256,0,0
130 BORDER 1,6: PAPER 0: INK 5
140 :
150 REPEAT game
160 REMark Set up a random display:
170 CLS: sc=RND(100 TO 500): sx=sc/1.5: sy=INT(sc/2)
180 SCALE sc,-sx,-sy: rd=sc/100: rn=INT(sy*.9)
190 FILL 1: CIRCLE 0,0,rd: FILL 0: x$=0: y$=0
200 x=RND(-rn TO rn): y=RND(-rn TO rn): ct=0
210 INK 2: FILL 1: CIRCLE x,y,rd: FILL 0: INK 7
220 :
230 AT 0,1: CLS 3: INPUT'Difficulty Level? (1 to 3)!'i$
240 IF i$='': GO TO 230
250 IF i$ INSTR'123456789': i=i$: ELSE GO TO 230
260 SElect ON i: =1:n=12: =2:n=9: =3:n=6
270 :
280 REPEAT loop
290 dist=INT(SQRT((x^2)+(y^2))): IF dist<15: NEXT game
300 ct=ct+1: AT 1,1: PRINT'Distance:!'dist,: d1=dist/n
310 :
320 REMark If last x-guess OK, dont input it again:
330 IF x<>x$ THEN
340 AT 2,1: INPUT'x?!'x$,
350 IF x$='': GO TO 340: ELSE IF x$=="q": NEXT game
360 :
370 FOR f=1 TO LEN(x$)
380 IF x$(f) INSTR'--+0123456789': ELSE GO TO 340
390 END FOR f
400 :
410 IF x$>=-dist AND x$<=dist: ELSE GO TO 340
420 END IF
430 :
440 REMark If last y-guess OK, dont input it again:
450 IF y<>y$ THEN
460 AT 3,1: INPUT'y?!'y$,
470 IF y$='': GO TO 460: ELSE IF y$=='q': NEXT game
480 :
490 FOR f=1 TO LEN(x$)
500 IF x$(f) INSTR'--+0123456789': ELSE GO TO 460
510 END FOR f
520 :
530 IF y$>=-dist AND y$<=dist: ELSE GO TO 460
540 END IF
550 :
560 REMark Show where the black-hole ends up:
570 OVER -1: INK 4
580 FOR f=1 TO 30: CIRCLE x$,y$,rd: i$=INKEY$(#1,8)
590 OVER 0: INK 2 : CIRCLE x,y,ct*d1: INK 7
600 :
610 IF ct=>n THEN
620 BEEP 12345,67: INK 0
630 FILL 1: CIRCLE 0,0,rd: FILL 0
640 INK 7: AT 6,1: PRINT'YOU LOST!'
650 AT 7,1: PRINT;'x'!x!!'y'!y: EXIT loop
660 END IF
670 :
680 IF x=x$ AND y=y$ THEN
690 BEEP 12345,6: INK 4
700 FILL 1: CIRCLE x,y,ct*d1: FILL 0
710 i$=INKEY$(#1,99)
720 INK 0 : FILL 1: CIRCLE x,y,ct*d1: FILL 0
730 INK 7: AT 6,1: PRINT'Bravo!!'ct!'tries'
740 EXIT loop
750 END IF
760 END REPEAT loop: i$=INKEY$(#1,300)
770 :
780 AT 9,1: PRINT'Another game? (y/n)'
790 i$=INKEY$(#1,-1): IF i$='': GO TO 780
800 IF i$=='y': ELSE EXIT game
810 END REPEAT game
820 WINDOW 256,206,256,0: SCALE 100,0,0: STOP

```

Byts of Wood

by Roy Wood

Is your little black box a joy and delight, a dust magnet or a doorstep? This is the question that has been exercising the minds of QL luminaries and humble key tappers over the last few months. Hardware or software? To emulate or not to emulate - that is the question. To press ALT/F1 - perchance to dream? And dreaming is what some people have been doing as spring made it's first foray into the Western Hemisphere and a young users fancy turned to thoughts of Super Gold Cards. After that piece of lyrical waxing - only marginally less painful than the tonsorial variety - we will move on.

QL USERS and ABUSERS

I always used to say that there were three types of QL Users:

Older Users who just want to use the old programs. They are happy with them and don't really want to have to learn anything new. Returning users who want to go back and play with a system they used years ago. Long time users who work at improving and using the system.

I am not saying this to, in any way, attach a spurious 'value' to any of the above categories. They are all part of the QL scene and they all offer and absorb something from the others. The categories are also not black and white and there are many areas of crossover where some users may have a foot in more than one camp depending on what they want to do at the time.

I have now had to add a fourth breed who have been emerging over the last few months. These are people who never owned a QL before and who want to get old systems and see what they can do with them. They want to enjoy the feel and appearance of the old 8 bit systems.

The whole reason that I mention this at this point is that there has been an ongoing hardware thread running through the QL Users internet list and the newer users have been the ones sparking this up. To some extent this has been the same old discussion we have had a few times before. So much so that one ex QL User, who suddenly started to receive the list again after signing off from it years back, suddenly found himself in Groundhog Day territory. He said he was reading the same discussions and arguments that he had been reading when he signed off the list and that there pretty much the same people there too. True, he stumbled in on the

perennial 'lets build more hardware/who going to do the drivers?/why is SMSQ/E not free?' thread that springs up like one of those Hammer Horror monsters when someone disturbs the ground, but I have to say that I enjoyed him saying that he found himself with 'tears of laughter' as he read it. It was making me chuckle a bit too and I was staying out of it.

Hardwaring

It did, however, shed a little light on a few things so I thought I would share my personal musings on the subject for this column.

Those people who have come back to the QL after a period of absence or who have found themselves washed up on our ebony beaches are willing to experiment and look around the QL world in ways many of our older users have given up. It did occur to me that, if our long term users still retained the enthusiasm that some of these people have we would have a much more active scene. Of course many of them lack the background knowledge of the way the QL works and which programs do what etc. but that is only to be expected. They are, to quote Heinlein 'Strangers in a Strange Land'.

One of the major questions is about hardware and the lack of it. There is a certain naivety about coming in from the modern world where there is so much cheap hardware available to a small cottage lit by the occasional candle and where the last actual hardware development was made 7 years ago - and that was not widely taken up. If you do not understand the problems involved in developing, prototyping and building hardware then it is hard to understand why an, otherwise active, society, such as ours, is not bristling with new developments.

When I came into active participation in the QL scene back, in 1995, it was just lulling into inactivity but there was a group of people who were taking up the challenge of bringing the system to life again. Nasta's Qubide gave us the first really reliable hard drive interface at the time and the Aurora really should have had more of an impact than it did. That prodded Tony Firshman and Lau Reeves into action and they came up with super-Hermes and it was all looking good. Unfortunately things came unravelled in the world of hardware soon after. By the time Stuart Honeyball

finally gave up on the idea of a Super Gold Card successor we were left with only Nasta who had any grand ideas or designs for the future and his personal circumstances left him high and dry at a crucial point in its design.

None of this is new to our long-standing users of course. What the newcomers and the returnees seem to be crying out for is a faster memory expansion with a hard disk interface on it and that is something which we almost had with the designs Nasta was demonstrating at the US shows a few years back.

Flash

In some ways, though, this is a dated concept. Flash memory is dirt cheap now. I just bought a 2Gb Secure digital Card for my IPAQ for £13. A Super Gold Card replacement with 2Gb of Flash RAM and a single 256Mb DIMM would be more than adequate for most people's needs. What is more it would have a faster access than most disk systems we have now. Finding a 68xxx CPU would be harder and writing the drivers and firmware needed to make all this happen would be difficult but not insurmountable. Of course that is supposing there were either a group of people or an individual willing to take it on and finance the project.

The real problem is the lack of people who would put their hands into their pockets to buy it. Back when the Q40 appeared it offered far better performance than any native hardware and most of the emulators we had at the time. The QL scene was bigger then but I only really got 15 takers willing to put money up front before the initial launch and that crept up to 25 by the time we started shipping it with a full blown colour SMSQ/E on board. I don't know how many D&D sold later but I suspect it was not dramatically better.

Tony Firshman wrote, recently, on the user group list about a new Super Gold Card replacement and suggested it would cost around £300. There were only three people then who jumped up to say they would take one (OK one person said he would buy two but that is still only 4 sales). A project like this would need sales in excess of 30 to break even and they would need to sell pretty fast before the burden of the purchase of the components placed too much financial strain on the originators.

New Head

There was a lot of discussion on the 'native

hardware' subject too. To some extent this has always puzzled me and took me back to an Eindhoven show a long time ago when Freddie Vacha of Digital Precision gave a talk. He said the best way to upgrade a QL was to 'Rip out the Gold Card, throw away the disk drives and run it native on microdrives'. Knowing Freddie's business acumen this was probably because he had a whole pile of microdrives in a box at home and he wanted to sell them. He certainly managed to clean up by selling a lot of people a whole swathe of outdated and largely unsupported software when he was making his exit from the QL scene.

Some of the people in the categories I mentioned above seem to have a similar attachment to the old black box itself and this should put them at odds with the faster, bigger, more powerful brigade. I do have a few original QLs sitting at home but I cannot say I feel any real need to hook them up to my ancient Philips monitor and do anything with Quill or Abacus. I will say, though, that Archive had a simplicity of use which many modern database programs lack, and it was capable of some quite interesting things if programmed - as Bill Cable's suite of programs demonstrated. Shame it was so good at destroying data too.

Some of the exponents of native hardware, however, are also among those asking for new expansions and faster CPUs. It all reminds me of the 'best broom I ever had - it has had two new heads and a new handle, but it is still good' joke. In the end I start to wonder if you got a small enough PC motherboard and slid it into the QL case, would this satisfy the need for the 'black box'? For a proportion of the 'native' fans the CPU itself is the main point and thrust of their attachment. They prefer to have access to the underlying 68xxx code but that code itself is not set in stone and the instruction sets vary greatly over the range. In some of the later 68xxx processors I believe it was even partially emulated. The latest versions of QPC2 have an emulation of the 68020 that is good enough to be a platform for George Gwilt's GWASS assembler so where is need for a real 68020 CPU?

So what is the point of this great attachment? Maybe one of its champions would like to write in to the magazine and explain it? It would have to be in lay terms so that the whole readership would benefit but it would be an interesting insight. I used to own an old 1952 Morris Minor car and was frowned upon by many of the car fanatics because I put a more up to date 1024cc

engine in place of the old 800cc side valve, but then, it was a car - I liked the way it looked but I wanted to arrive somewhere before I died of old age.

UMPC!

No - that is not a comment - it stands for Ultra Mobile PC. Having mentioned tiny PC motherboards I should maybe talk about these little beasts. The UMPC format was announced over a year ago but the computers themselves are only just hitting the shops. The interesting thing here is that they are halfway between a PDA and a Tablet PC - about the size of a large paperback but quite slim. There is a page of general UMPC specs here:

<http://www.intel.com/design/mobile/platform/downloads/umpc2006.pdf>

The unit has an 800 x 480 7" TFT screen and runs versions of Windoze (I know) but that does mean you could have QPC2 running on it if the operating system has not been changed too much to fit the device. The Samsung version has provision for USB keyboard although I am not sure about how the touch screen would work as a mouse but, in theory, it could be the ultimate QL in your pocket.

Some years ago, before QBranch, I wrote in to Quanta about trying to make a portable QL. I wanted to be able to take my QL out on the road with me when I was on tour. I later achieved this with an old 286 laptop and a QXL card but this new format could really be the business.

Having said that they are not cheap devices but that is because they are the latest 'must have' thing for techno geeks. After they have fleeced the first few thousand users the price will tumble.

Those Benchmark Blues

Someone - I forget who right now - also mentioned that his QPC2 running on a Core2 Duo Intel CPU was slightly slower than his Q60. Now I am not sure what relevance this has. No QL program that I can think of needs Grand Prix (to continue the Top Gear motoring analogy) performance. I know we have a few long winded processes like POV (Persistence of Vision) and other software that takes it's time producing things but, if you have a fast PC to run QPC2 on, why do do this stuff in faster more up to date programs? I actually find the increased speed annoying at times when a single mouse click opens two versions of the program I was trying to run.

From Behind the Screen

Just as I was running towards the end of this column I received an email from an anonymous user saying he would not be re-subscribing to QL Today because of Geoff Wick's attitude towards Quanta. I had mixed feelings about this kind of comment and, since I was the recipient of some similar quotes a while back, I would like to make a few comments here and add my own opinion.

Email is a continuing problem for exchanges like this and the roots of the apparent schism between Geoff and the Quanta committee lie there. I know, more than most probably, that, when you sit in front of a screen reading a text and you see a loophole or a chink in the armour of your opponent, you go 'Yes' and the fingers fly followed by the 'Send' key. Maybe later you think 'maybe that sounded harsher than I meant it to be' but by that time it is buzzing its way through the copper like an electron tipped dart. Once a few of these emails have been exchanged, open war can easily commence and, when you look at this from the outside it can seem a lot harder than it actually is.

I spend a bit of time poking fun at Dilwyn and Tony Firshman both in this column and on the Users list but it is fun. I am not serious even though some people seem to have the impression I am and ask me to 'back off'. Sometimes I even get people who take these little jibes very seriously and start to foam at the mouth and complain in somewhat vitriolic terms. You cannot hear the inflection in an email (or indeed on the printed page) or see the expression on the face of the writer so it is easy to get it wrong.

At the heart of all of this, though, is a different matter. I cannot speak for Geoff, and I am sure he will be speaking for himself elsewhere in the magazine, but I am of the opinion that he finds Quanta rather frustrating and feels it could do better. Quanta has done some great things in the past. It has rescued software and publications, sponsored shows, ran an extensive library and produced the QIMI mouse and a run of Super Gold Cards when the original producer wanted out and sponsoring the Q60. Like all organisations there are times when it rests on its laurels and like all organisations it sometimes needs a swift application of the pedial appendage to its fundamental orifice to get it to realise The laurels have wilted and it is time pick some more.

Right now we have a few problems in QLworld. There is a need for new hardware and a need for innovation. there are precious few people out

there who can satisfy that need and Quanta should be in forefront of trying to energise and motivate people to do this. Sometime ago, when Quanta had a batch of Super Gold Cards made there was a furious row at an AGM with Dr Phil Jones (remember him? He made Geoff look like a poodle) suggested that this what not what Quanta was about but he was wrong. It is exactly what it should be about. There is no point sitting on a pot of gold until it becomes worthless, it should be used for the benefit of the people who contributed it. Quanta's attitude is that it is waiting for people to come forward with ideas and a business plan but it should be going out there and persuading people to do this work. In my scuba diving club there are times when I have tried to organise events and put out a general message asking for instructors to help out or experienced people to come and run a dive. Talking to some of these later they often say 'oh, I would have come, but you were just asking for volunteers and I was sure someone else would do it. You should have asked me.'. Let's face it to someone else we are all someone else. Our club could have a whole lot more money if we just collected the member's fees, sold the boats and the other kit and sat in the pub talking about diving. It is a dive club because we go diving and this is a QL club by all but name and it should have a similarly succinct agenda.

At a point in time where there is a definitive need and one which, in part, could be tackled by financial means, Quanta, which has the resources, should be actively investigating that need and not waiting for a solution to come up to its door with a business plan and a begging bowl. If not for its own members then for its own survival.

Just Words

You could accuse Geoff just sitting there making smug comments. Why is he not doing anything? He did try to get elected to the committee a while back and I am sure he would be trying to do all this if he had been elected. None of the foregoing is meant to, in any way, belittle the work done by key members of Quanta, many of whom have turned out tirelessly for every show. It may just be that, in turning out so diligently, they have lost some of the spirit of enterprise that we now need so dearly.

Now you might say that this against the constitution that Quanta has to abide by but then a constitution is just a bunch of words on a piece of paper. You can write a new one and then get that voted in. It may be that Quanta can only sell

its goods to Quanta members but, here a thought, non members could buy any new QL peripheral for a bit more than members could but would get a free year's subscription to it. Might even boost the membership.

Anyway back to less contentious matters.

Here is a little conundrum for you.

I have a zip file which was sent to me by Jim Hunkins with QDT v 1.01 on it. It has sat on my downloads directory for ages since v1.02 is the current one. I was recently asked for a copy of it because the user had only v1.00 and you have to go through v1.01 to upgrade to v1.02.

The zip file is the one I used to create the distribution of QDT that I sold for some time with complete success and no problems from customers. The archive contains three files:

```
qdtinst.bas
QDTINST.zip
unzip
```

OK. So far so good. I copied the whole archive to RAM2_ and unzipped it, using the QL Unzip program, to RAM1_ I then copied the three files to flp1_ and sent it out.

When my customer came to use it he found he got an error message about the zip file having no end of file signature and called me to tell me. I repeated the first part of the above procedure and checked the QDTINST.zip file. This unzipped OK. I copied it to a floppy drive and it failed.

I tried this several times, using a new floppy disk, using different methods of copying the file but always the same result. Finally I unzipped the original archive direct from the PC side to a QL floppy (Good thing QPC2 can see the PC hard drive so easily) and that works.

Does anyone have any ideas on why this should be?

And finally....

Reading back through this column I realise that I have been far more serious and there was not the amount of puerile silliness that I am known for. In order to redress that balance a little I would like to give you a joke for all the maths geeks out there.

Question: Why do maths geeks get Halloween and Christmas confused?

Answer: Because DEC 25 is the same as OCT 31 (You may need to ask a mathematician about this.)

The QL Show Agenda

QL Meeting in Eindhoven

Saturday, 20th of October, 10:00 to 16:00

Pleincollege St. Joris, Roostenlaan 296

Thanks to the organiser, Sjef van de Molengraaf, the meetings at Eindhoven continue. Same venue as always. J-M-S will be there, as always. Roy Wood of QBranch also plans to attend this show. Details in the next issue.

Just (a few) Words!

If you carefully read the ads in this issue of QL Today, you will have read that our trusty Editor, Geoff Wicks of Just Words! has provided us with his last advertisement. We, the other "makers" of QL Today, would like to thank Geoff for his dedication to help producing the contents of this magazine.

We are very glad that he jumped in when Dilwyn had to stop being the Editor ... and we are also very glad that Dilwyn manages to continue supporting QL Today. A big "THANK YOU" also to you, the readers and authors of articles and all therefore all the supporters of QL Today! We, the team, look forward to another volume of QL Today.

The Next Volume

Please fill in and return the renewal form enclosed as soon as possible, to help in saving costs.
Thanks to everybody who returned the renewal form already - thanks for you support!

German subscribers with automatic renewal and account debit do not need to return anything, the subscription will be automaticall renewed as agreed.

The Next Issue

We plan to have the next issue ready for you in the middle of September. As always, it depends on how quickly we will get reviews, articles etc. from you. Please send in material a.s.a.p.